ESKORT Clearance Risk Analysis 3.1

Clearance Workbench User Guide

icdk/ECL/CWB/User Guide 3.1/v1.2/2011.11.15
Copyright © 2009 Intracom IT Services Denmark A/S. All rights reserved. Commercial In Confidence

DOCUMENT

Title:	Clearance Workbench User Guide
Document:	icdk/ECL/CWB/User Guide 3.1
Date:	2011.11.15
Version:	1.2
Author:	Staffan Anderberg, Jørgen Mortensen
Contributions by:	Don Cameron, Robert Merineau, Christoffer Børrild
Classification:	None
Distribution:	Project

Versions:

1.0	Initial release of ESKORT Clearance Risk Analysis version 3.1
1.1	Updated documentation of CountToWithKey-function
1.2	Updated the section “The Autocomplete feature”

Printed:

Table of Contents

Preface 6

1.1 Notation Conventions 6
1.2 Mouse Operation Conventions 7
1.3 Highlighting (Selecting) Lines 7
1.4 Note on Conformity 7

2. Introduction 8

2.1 Modules in the ESKORT Clearance Risk Analysis System 8
2.2 The Risk Analysis Process 8
2.3 The Risk Analysis Configuration 8
2.4 Revisions 9

3. User Interface 10

3.1 Screen Layout 10
3.1.1 Re-sizing the Main Areas 11
3.1.1.1 Using the Dividing Bars to Re-size the Main Areas 11
3.1.1.2 Using the Re-sizing arrows to Re-size the Main Areas 12
3.2 The Main Menus 12
3.2.1 How to Display the Menu 12
3.2.2 How to Display Menu Commands 12
3.2.3 The File Menu 13
3.2.4 The Edit Menu 14
3.2.5 The Help Menu 15
3.3 The Toolbar 16
3.4 The Main Areas 18
3.4.1 The Filter Area 18
3.4.1.1 Executing Filtering Operations 20
3.4.2 The Table Area 20
3.4.2.1 Selecting and Opening Configuration Elements 21
3.4.2.2 Duplicating Configuration Elements 22
3.4.2.3 Deleting Configuration Elements 22
3.4.2.4 Printing Configuration Elements 22
3.4.2.5 Moving and Changing Columns 24
3.4.2.6 Sorting Table Area Contents 25
3.4.2.7 Hiding and Displaying Columns 27
3.4.3 The Data Area 27
3.4.3.1 Displaying Configuration Elements 28
3.4.3.2 Closing Configuration Elements 28

4. General Configuration Items 29

4.1 Declaration Types 29
4.2 Declaration Segments 31
4.3 Date and Time 32
4.4 Customs Office 32
4.5 Keyword 32
4.6 Trader Number 33

4.7 User ***33***
4.8 Hyperlinks ***33***
4.8.1 Create a Hyperlink 34
4.8.2 Edit a Hyperlink 34
4.8.3 Open a hyperlink 34
4.9 Wildcards ***35***

5. Working with Risk Profiles 36

5.1 Define a new Risk Profile ***37***
5.1.1 The Header sub-tab 37
5.1.2 The Schedule sub-tab 38
5.1.2.1 General Activation and Suspension 38
5.1.2.2 Working with Recurrence Patterns 45
5.1.3 The Declaration Types sub-tab 54
5.1.4 The Description sub-tab 55
5.1.5 The Risk Profile sub-tab 56
5.1.6 The Control Request sub-tab 56
5.1.6.1 Selecting Inspection Type 57
5.1.6.2 Examination Instructions 57
5.1.6.3 Selecting Report Back Fields 58
5.1.7 Saving the Risk Profile 59
5.2 Editing a Risk Profile ***59***
5.3 Filtering Risk Profiles ***59***
5.3.1 Filtering using the State Group 60
5.3.2 Filtering using the Risk Profile ID Number 61
5.4 Duplicating Risk Profiles ***61***
5.5 Risk Profile Revisions ***61***
5.6 Printing ***62***
5.7 Working with Risk Profiles Rules ***62***
5.7.1 Re-sizing the Views 63
5.7.2 Using the Dividing Bars to Re-size the Views 63
5.7.3 Using the Re-sizing arrows to Re-size the Views 63
5.7.4 Using the Display Buttons to Re-size the Views 64
5.7.5 Rule Buttons 64
5.7.6 The Partial Expression View 65
5.7.6.1 Columns 65
5.7.6.2 Moving and Changing Columns 66
5.7.6.3 Setting Conditions 66
5.7.6.4 Deleting, Undoing and Redoing Conditions 68
5.7.7 The Formalized Expression View 68
5.7.7.1 Writing Conditions 69
5.7.7.2 The Auto Complete feature 69
5.7.7.3 Using Aggregated Data 70
5.7.7.4 Deleting, Undoing and Redoing Conditions 84
5.7.7.5 Validation of Rules 84
5.7.8 Indicators 84
5.7.9 Operators 85
5.8 Testing Risk Profiles ***85***
5.8.1 Background 85
5.8.2 General Concept – Test Task 86
5.8.3 Results 86
5.8.4 Optimizing Risk Profiles 87
5.8.4.1 Create an Initial Risk Profile 87
5.8.4.2 Define a Test Task 88
5.8.4.3 Create Test Data (Declarations) 88

5.8.4.4 Executing the Test Task 88
5.8.4.5 Examine the Results 88
5.8.5 Working in the Testbed 91
5.8.5.1 Test Tasks in Risk Profiles 91
5.8.5.2 Test Scenarios 98

6. Working with the Compliant List 102

6.1 Add a Trader to the Compliant List 102
6.2 Edit a Trader in the Compliant List 103
6.3 Delete a Trader from the Compliant List 104
6.4 Duplicate a Trader in the Compliant List 104
6.5 Printing 104

7. Working with Result Tables 105

7.1 Create a new Result Table 105
7.2 Editing a Result Table 106
7.3 Deleting a Result Table 106
7.4 Duplicate Result Tables 106
7.5 Printing 107

8. Working with Risk Objects 108

8.1 Creating new Risk Objects 108
8.1.1 Creating new Procedure, Trader and Commodity Risk Objects 108
8.1.2 Creating a new Country Risk Object 109
8.2 Editing a Risk Object 110
8.3 Deleting a Risk Object 110
8.4 Duplicating Risk Objects 111
8.5 Printing 111

9. Working with Random Selections 112

9.1 Create new Random Selection Configuration Elements 112
9.2 Edit Random Selection Configuration Elements 112
9.3 Deleting Random Selection Configuration Elements 113
9.4 Duplicating Random Selection Configuration Elements 113
9.5 Printing 113

10. Data Loading Tool 114

10.1 Step 1 – Starting the Data Loading Tool 114
10.2 Step 2 – Specifying the Source File 115
10.3 Step 3 – Configuring the Interpretation of the Source File 117
10.3.1 Delimiters 117
10.3.2 Consecutive Delimiters 117
10.3.3 Text Qualifiers 118
10.3.4 Text Qualifiers Used in the Text 118
10.3.5 Start Import at Row 119
10.4 Step 4 – Mapping up the Columns 120
10.4.1 Data Mapping 120
10.4.2 Source Format 120
10.4.2.1 Date Type 120
10.4.2.2 Numeric Types 122

10.4.3 Data Preview 123
10.5 Step 5 – Starting the Import..... 124
10.6 Step 6 – Import Reporting 125
10.7 Step 7 – After the Import 126

11. Appendix 1: MDC Language Definition..... 128

11.1 Grammar..... 128
11.2 Terminal Symbols 133

12. Appendix 2: Function Reference..... 134

12.1 Special argument types..... 134
12.1.1 TUPLE (a.k.a. COORDINATE) 134
12.1.2 ANY_TYPE (a.k.a UNKNOWN) 134
12.2 Functions..... 135
12.2.1 AbsValue (double x) 135
12.2.2 ActivateArea (string AreaName) 135
12.2.3 AllEmpty (TUPLE Measure, string DimensionName)..... 136
12.2.4 AllZeroOrNotKnown (TUPLE Measure)..... 136
12.2.5 AnyZeroOrNotKnown (TUPLE Measure) 137
12.2.6 Assign (TUPLE Tuple, ANY_TYPE Value)..... 137
12.2.7 Average (TUPLE Measure, string DimensionName)..... 138
12.2.8 BuildDate (ANY_TYPE date, string modification) 138
12.2.9 ConcatStr (string s_1, string s_2)..... 139
12.2.10 CountTo (integer n)..... 139
12.2.11 CountToWithKey (integer n, string key) 140
12.2.12 CubeEmpty (string CubeName)..... 140
12.2.13 DateHalfAYearBack (string date)..... 141
12.2.14 DateToLong (string date) 141
12.2.15 DateXYearBack (string date, double nYears)..... 142
12.2.16 DaysBetween (string startDate, string endDate) 142
12.2.17 DaysPassed (string since) 143
12.2.18 DeActivateArea (string AreaName)..... 143
12.2.19 DblToLng (double x) 143
12.2.20 DblToStr (double x)..... 144
12.2.21 Difference (double x1, double x2) 144
12.2.22 EndDate (string period)..... 144
12.2.23 ExecuteArea (string AreaName)..... 145
12.2.24 FilterString (string s, string trimChars) 145
12.2.25 FormatNumber (double value, string format) 145
12.2.26 GetCubeName (CURRENT_CUBE) 146
12.2.27 GetNoOfMembers (CURRENT_CUBE, string dimensionName) 146
12.2.28 GetParamDoubleValue(string UsageName)..... 147
12.2.29 GetParamIntValue (string UsageName)..... 147
12.2.30 GetParamStrValue (string UsageName)..... 147
12.2.31 GetRandomNo (double min, double max)..... 148
12.2.32 InString (string candidate, string s)..... 148
12.2.33 IsEmpty (TUPLE Measure)..... 149
12.2.34 IsKnown (TUPLE Measure)..... 149
12.2.35 IsKnownAndNotZero (TUPLE Measure)..... 150
12.2.36 IsNotKnown (TUPLE Measure) 150
12.2.37 IsStandardContainerCode (string code)..... 151
12.2.38 IsZeroOrNotKnown (TUPLE Measure)..... 153
12.2.39 LeftStr (string s, double length) 153
12.2.40 LngToDbl (long value)..... 154

12.2.41 LngToStr (long value) 154
12.2.42 LowerCase (string s) 154
12.2.43 Maximum (double x_1, double x_2) 155
12.2.44 MaximumValue (TUPLE Measure, string DimensionName) 155
12.2.45 Minimum (double x_1, double x_2) 155
12.2.46 MinimumValue (TUPLE Measure, string DimensionName) 156
12.2.47 OnChanged (string key) 156
12.2.48 p (string ParameterTableName) 157
12.2.49 pa (string ParameterTableName) 157
12.2.50 py (string ParameterTableName, string Year) 157
12.2.51 QualifiedAverage (TUPLE Measure, double Threshold, string DimensionName) 158
12.2.52 QualifiedCount (TUPLE Measure, double Threshold, string DimensionName) 158
12.2.53 randomsample (double rate) 159
12.2.54 Ratio (double arg_1, double arg_2, double arg_3, double arg_4) 159
12.2.55 RatioChange(TUPLE numerator, TUPLE denom, long year) 160
12.2.56 RatioDrop (TUPLE numerator, TUPLE denom, long year) 160
12.2.57 RatioIncrease(TUPLE numerator, TUPLE denom, long year) 161
12.2.58 RightStr (string s, double length) 161
12.2.59 Roundsum (double value) 162
12.2.60 SoundsLike (string s_1, string s_2) 162
12.2.61 StartDate (string period) 163
12.2.62 StrToDbl (string s) 163
12.2.63 StrToLng (string s) 163
12.2.64 SubStr (string s, double pos, double length) 164
12.2.65 Today () 164
12.2.66 TotalDeviation (TUPLE Measure) 164
12.2.67 TotalDeviationPct (TUPLE Measure) 165
12.2.68 TrimString (string s) 165
12.2.69 UpperCase (string s) 165
12.2.70 ValidAirWaybillNumber(string awbn) 166
12.3 Reference sections 167
12.3.1 Numeric Display Formats 167
12.3.1.1 Precision Flag 168
12.3.1.2 Digit Grouping 168
12.3.1.3 Decimal Point Character 169
12.3.1.4 Scientific notation 169
12.3.1.5 Force or remove sign 169
12.3.1.6 Trailing zeros 170
12.3.1.7 Leading zeros 170
12.3.1.8 Multiplication 170
12.3.1.9 Formal Syntax 170
12.3.1.10 The Default Settings 171
12.3.1.11 Examples 171

°LastInfoPage

Preface

1.1 Notation Conventions

Within this user guide the following notation conventions are followed:

- Application names are written in italic bold face, e.g. ***Clearance Workbench*** and ***Designer***.

- Main user interface parts are written in capital letters, e.g. Toolbar, Data Area and Menu.
- User interface elements are written in bold face, such as references to e.g. **menu items**, **dialog** and **window** names, **buttons** and **text boxes**.
- Keyboard key references are written in uppercase letters, e.g. SHIFT and CTRL.

1.2 Mouse Operation Conventions

The user guide adheres to the following conventions regarding using the mouse:

- Click: press and release the primary mouse button.
- Double click: press and release the primary mouse button twice in quick succession.
- Right click: press and release the secondary mouse button.
- Drag: press the primary mouse button, move it to another control, dragging something with it, and then release.

☞ IMPORTANT: Left hand and right handed mice: If your mouse is set up for right handed use, then the primary mouse button is on the left, and the secondary mouse button is on the right. That is, you "right click" with the right button. If your mouse is set up for left handed use, then the buttons are reversed: you "right click" with the left button on the mouse.

1.3 Highlighting (Selecting) Lines

Throughout the user guide it is described that you can highlight (select) lines. The ***Clearance Workbench*** offers the following key combinations together with mouse clicking to highlight (select) lines:

- A mouse click highlights (selects) a single line, overriding any already highlighted (selected) lines.
- A mouse click combined with pressing the SHIFT key highlights (selects) consecutive lines from the last selection.
- A mouse click combined with pressing the CTRL key selects a single line while retaining the previous selection

1.4 Note on Conformity

This user guide describes and corresponds to the standard version of the ESKORT Clearance Workbench. The system can however be reconfigured with the use of the **ESKORT Designer** and it should thus be noted that due to such reconfiguration this user guide may not reflect such changes having been made.

Due to the fact that in the different installations of the standard system some terms like e.g. risk profiles/trap rules and Started at/Activated at are interchangeable there may be some discrepancies between the figures in this guide and the corresponding screens in some customer systems.

2. Introduction

2.1 Modules in the ESKORT Clearance Risk Analysis System

Three modules, the ***Analysis Server***, the ***Clearance Workbench*** and the ***Designer***, comprise the ESKORT Clearance Risk Analysis system. The ***Analysis Server*** is the application that performs the risk analysis of incoming customs declarations and produce analysis results that are communicated to the declaration processing system for further handling. The ***Clearance Workbench*** and the ***Designer*** are tools for administering the Risk Analysis Configuration, which is the interface between the two administrative modules and the ***Analysis Server.*** The Risk Analysis Configuration is continuously used by the ***Analysis Server*** to assure that, at any time, the analysis is based on the latest available information. The day-to-day updating of the rules and parameters is carried out in the ***Clearance Workbench*** and this task is executed by several customs officers placed both centrally and regionally/locally. The ***Designer*** however, is a tool for more substantial long term changes of the risk configuration (such as introducing new declaration fields, new algorithms for calculating risk values) and this work is carried out by dedicated super users.

This user guide covers the work that is carried out in the ***Clearance Workbench***.

2.2 The Risk Analysis Process

The risk analysis process is performed in three separate phases in the ESKORT Clearance Risk Analysis system. In the first phase (Phase 1) risk profiles are used to trap high risk shipments and recommend a compliance action such as physical inspection of the goods or document inspection. In the second phase (Phase 2) risk scores are calculated for the declarations and they are then classified into risk groups in order to focus the attention on the high risk groups. In the third phase (Phase 3) the declarations are exposed to a random selection where flexible sampling criteria is used for selecting declarations for inspections.

All declarations pass through Phase 1 and Phase 2 and those that are not picked for inspection in these phases will also pass through Phase 3.

2.3 The Risk Analysis Configuration

All rules, conditions and parameters defined and updated in the ***Clearance Workbench*** are stored in the Risk Analysis Configuration as configuration elements. Configuration elements are the various entities that exist and/or are created inside the ***Clearance Workbench***, such as individual risk objects and risk profiles. The configuration elements are divided into five groups:

- Risk Profiles (also known as Trap Rules in some installations)
- Compliant List (also known as Blockings in some installations)
- Result Tables
- Risk Objects
- Random Selection (also known as Random Sampling in some installations)

2.4 Revisions

A configuration element can exist in several revisions (or versions). When e.g. a risk profile is created and saved it will be assigned as the first revision and be allotted revision No. 0. If later the same risk profile is updated it will then be stored as a new revision, this time with revision No. 1. For all configuration elements except risk profiles only the latest revision is available in the ***Clearance Workbench*** but the previous revisions are stored in the risk analysis database. For risk profiles all older revisions can be opened for use, refer to section 5.5 Risk Profile Revisions.

3. User Interface

3.1 Screen Layout

The user interface in the ***Clearance Workbench*** reflects the three phases in the risk analysis process. Thus, there are five Main Tabs that you can select in order to update the analysis configuration:

Main Tab Name	Risk Analysis Phase
Risk Profiles Tab	Phase 1
Compliant List Tab	Phase 1-3
Result Tables Tab	Phase 2
Risk Objects Tab	Phase 2
Random Selections Tab	Phase 3

Access to the various tabs is determined by the roles of the users and depending on your user role you may not have access to all five main tabs and/or you may have limited access (e.g. read-only access).

Each main tab adheres to the same overall layout and consists of three main areas as illustrated by the Risk Profiles Tab below. The main areas are the Filter Area, the Table Area and the Data Area.

The main areas are separated by **dividing bars** and each **dividing bar** has a set of **re-sizing arrows**.

The screen also contains a Menu bar in standard Windows format and a Toolbar for easy and quick access to frequently used functions.

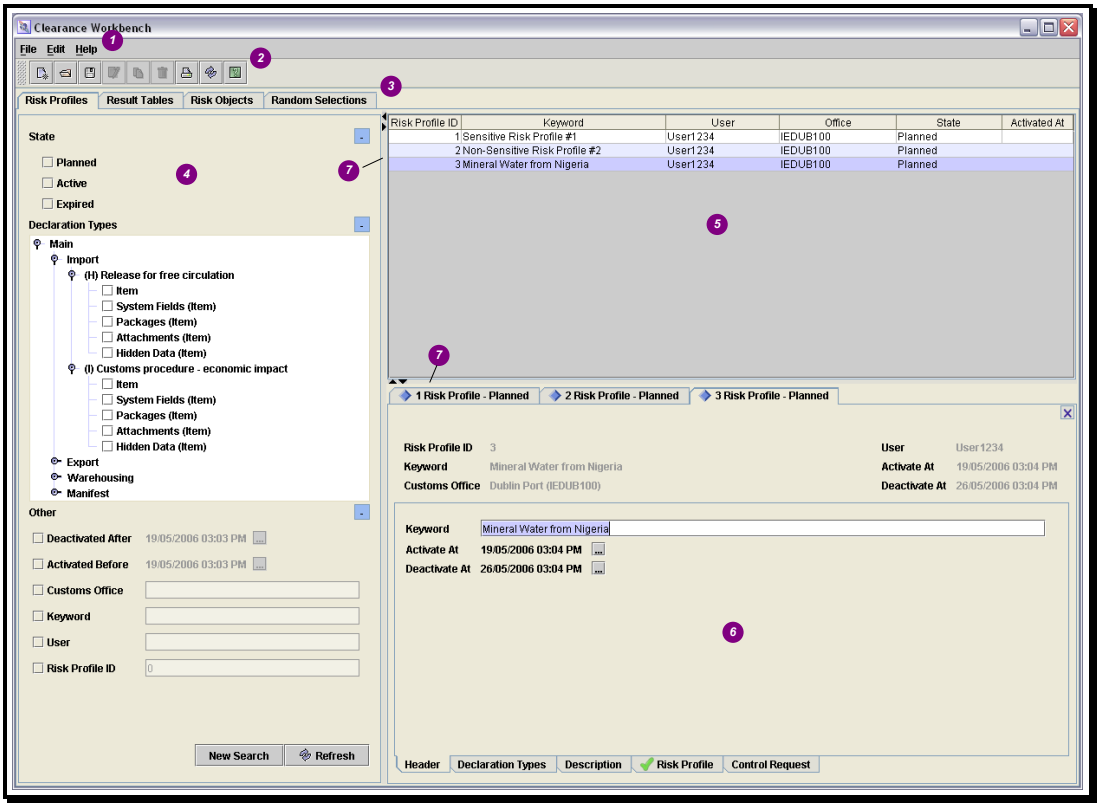


Figure 1: Overall Screen layout

❶ Menu Bar

❷ Toolbar

❸ Main Tabs

❹ Filter Area

❺ Table Area

❻ Data Area

❼ Dividing Bars with Re-sizing Arrows

3.1.1 Re-sizing the Main Areas

Each of the main areas can be re-sized whenever necessary, e.g. to display more columns in the Table Area. There are two methods for use; dragging the **dividing bars** and clicking the **re-sizing arrows**.

3.1.1.1 Using the Dividing Bars to Re-size the Main Areas

To re-size a main area do as follows:

1. Move the cursor across one of the **dividing bars**

2. When the cursor changes shape to a double-pointed arrow
   drag the **dividing bar** in the desired direction

3.1.1.2 Using the Re-sizing arrows to Re-size the Main Areas

To re-size a main area do as follows:

1. To maximize a main area click the **re-sizing arrow** that is pointing away from the area
2. To restore the size to normal click the **re-sizing arrow** that is pointing towards the area
3. To minimize a main area click the **re-sizing arrow** that is pointing towards the area
4. To restore the size to normal click the **re-sizing arrow** that is pointing away from the area

3.2 The Main Menus

The ***Clearance Workbench*** Menu bar shown below is a Windows standard menu.

3.2.1 How to Display the Menu

To display a menu, click with the mouse on the title in the Menu bar. Alternatively press ALT followed by the underlined letter in the menu. To e.g. open the **File** menu press ALT-F.

☞ Note: The underlined letter in the menu items is the access key for the menu command, which can be used in combination with the ALT key.

3.2.2 How to Display Menu Commands

To select a specific command, click with the mouse on its title in the drop down menu. Some common commands have a shortcut key combination such as e.g. the **File** menu above where you can press CTRL-O to open a configuration element.

3.2.3 The File Menu

The **File** menu contains the following commands:

File Menu Item	Shortcut Key	Equivalent Toolbar Button	Command	Description
New	**CTRL-N**		Create new configuration element	Creates a new risk profile, a compliant list trader, a result table, a risk object or a random selection
Open	**CTRL-O**		Open configuration element	Opens a risk profile, a compliant list trader, a result table, a risk object or a random selection in read-only mode
Save	**CTRL-S**		Save configuration element	Saves a risk profile, a compliant list trader, a result table, a risk object or a random selection after edit
Exit	**ALT-F4**	None available	Exit application	Exits the ***Clearance Workbench*** application

3.2.4 The Edit Menu

The **Edit** menu contains the following commands:

Edit Menu Item	Shortcut Key	Equivalent Toolbar Button	Command	Description
Edit	**CTRL-E**		Edit configuration element	Opens a risk profile, a compliant list trader, a result table, a risk object or a random selection for edit
Duplicate	**CTRL-SHIFT--D**		Duplicate configuration element	Duplicates a risk profile, a compliant list trader, a result table, a risk object or a random selection and opens it for edit
Show Revisions	**CTRL-R**		Activates the revisions function for risk profiles	Opens a risk profile in a new tab to allow working with revisions
Delete	**DELETE**		Delete configuration element	Deletes a risk profile, a compliant list trader, a result table, a risk object or a random selection
Print	**CTRL-P**		Print Table Area and configuration elements	Prints lists and details of configuration elements

Edit Menu Item	Shortcut Key	Equivalent Toolbar Button	Command	Description
Refresh	**F5**		Refresh configuration elements	Searches the ESKORT Database for risk profiles, compliant list traders, result tables, risk objects or a random selections according to the current filter settings and list them in the Table Area

3.2.5 The Help Menu

The **Help** menu contains the following commands:

Help Menu Item	Shortcut Key	Equivalent Toolbar Button	Command	Description
Help	**F1**		Open on-line Help	Opens the on-line Help system in a new window
Help Contents	**SHIFT-F1**	None available	Open on-line Help contents	Opens the contents in the on-line Help system in a new window
About	None available	None available	Open About box	Opens the About box with relevant information about the ***Clearance Workbench***

3.3 The Toolbar

The Toolbar in the ***Clearance Workbench*** contains buttons that enable quick access to commands.

Toolbar button	Shortcut Key	Equivalent Menu Item	Command	Description
	CTRL-N	File Menu, New	Create new configuration element	Creates a new configuration element
	CTRL-O	File Menu, Open	Open configuration element	Opens a risk profile, a compliant list trader, a result table, a risk object or a random selection in read-only mode
	CTRL-S	File Menu, Save	Save configuration element	Saves a risk profile, a compliant list trader, a result table, a risk object or a random selection after edit
	CTRL-E	Edit Menu, Edit	Edit configuration element	Opens a risk profile, a compliant list trader, a result table, a risk object or a random selection for edit
	CTRL-SHIFT-D	Edit Menu, Duplicate	Duplicate configuration element	Duplicates a risk profile, a compliant list trader, a result table, a risk object or a random selection and opens it for edit
	Delete	Edit Menu, Delete	Delete configuration element	Deletes a risk profile, a compliant list trader, a result table, a risk object or a random selection
	CTRL-P	Edit Menu, Print	Print Table Area and configuration elements	Prints lists and details of configuration elements
	F5	Edit Menu, Refresh	Refresh the Table Area	Searches the ESKORT Database for risk profiles, compliant list traders, result tables, risk objects or random selections according to the current filter settings and list them in the Table Area

Toolbar button	Shortcut Key	Equivalent Menu Item	Command	Description
	F1	Help Menu, Help	Open on-line Help	Opens the on-line Help system in a new window
	CTRL-R	Edit Menu, Show Revisions	Activates the revisions function for risk profiles	Opens a risk profile in a new tab to allow working with revisions

3.4 The Main Areas

All the five tabs in the ***Clearance Workbench*** have the same basic layout with a Filter Area, a Table Area and a Data Area. The detailed contents of these main areas differ however between the Main Tabs. This section contains a description of the main areas from a general point of view while the specific functionalities in each main area are described in the sections describing each tab.

3.4.1 The Filter Area

Working in the ***Clearance Workbench*** you often need to locate previously defined configuration elements in order to change, update or refine them. The system contains several thousands of configuration elements, in particular Risk Objects, and in order to avoid having to scroll through extensive lists of configuration elements you can use the options in the Filter Area. In the Filter Area you can define criteria for searching/filtering the configuration elements you wish to work with. Examples of such criteria are declaration types and customs office codes.

Inside the Risk Profiles Filter Area (see Figure 2 below) the criteria is divided into three groups; **State**, **Declaration types** and **Other**. For the Compliant List tab there are two groups – **Search Conditions** and **Other** - but for the other tabs there is only one group in each tab. The groups have **expand/collapse** buttons that can be used for expanding the criteria inside a group or collapse the criteria. An **expand/collapse** button can vary between two states depending on whether the group is expanded or collapsed; if a group is collapsed the button contains a plus sign + and if a group is expanded it contains a minus sign -. By clicking these buttons the contents in a group can be expanded or collapsed.

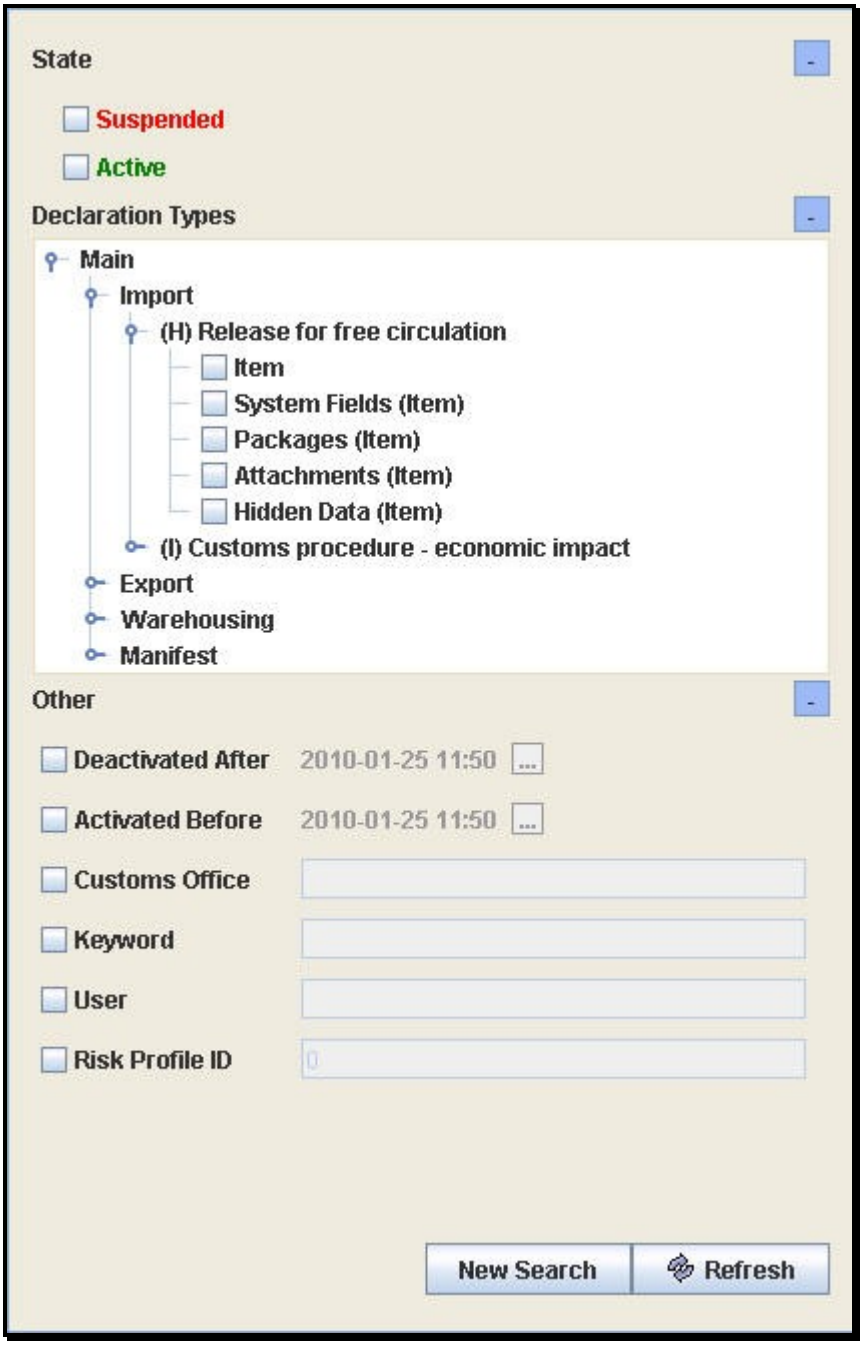


Figure 2: Filter Area for risk profile tab

Inside the **Declaration Types** group there are four sub-groups; **Import, Export, Warehousing** and **Manifest**. Each of these sub-groups contains a number of declaration types and they have **expand/collapse** icons that can be used for expanding or collapsing the sub-groups. An **expand/collapse** icon can vary between two states depending on whether the sub-group is expanded or collapsed; if a sub-group is collapsed the **expand** icon, , can be used for expanding the sub-group and if a sub-group is expanded the **collapse** icon, , can be used for collapsing the sub-group.

For each declaration type there is a number of declaration segments and you can select one or several of them by ticking the tick boxes.

In the Risk Objects tab filter area there is a drop-down box at the top for selecting the risk object type for which you wish to work with. The filter criteria that are available differ between the various risk object types.

3.4.1.1 Executing Filtering Operations

A filtering operation is executed in the same way in all tabs - except the Risk Objects tab - although the filter criteria differ among the tabs. For the Risk Objects tab you first have to select risk object type from the Risk Object drop-down box. In order to perform a filtering operation you enter the relevant criteria by ticking the relevant tick box(es) and/or entering the desired values into the text boxes.

☞ **Note:** For tick boxes that have a text box to the right of the tick box you can click in the text box and as a result the tick box is ticked and the text box is opened for edit.

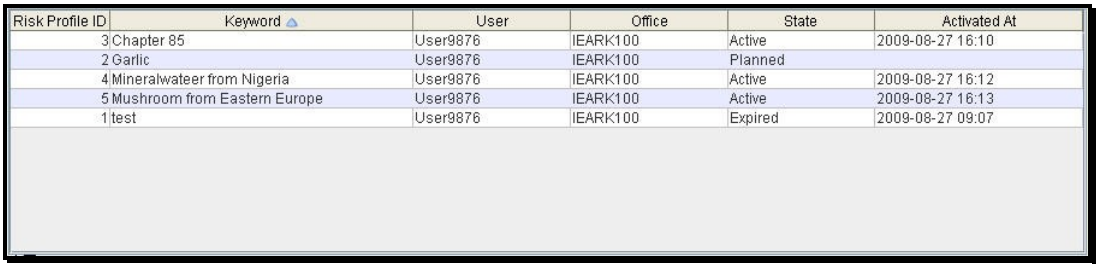
When you have entered the desired criteria you click the **Refresh** button in the lower right hand corner of the Filter Area. Alternatively you can click the **Refresh** button, , on the Toolbar, select **Refresh** from the Edit Menu or press **F5**. As a result the configuration elements that match the filter criteria will be listed in the Table Area for further work. If the list still is too extensive you can refine the filter criteria by changing values or adding criteria and then refresh again whereby the list in the Table Area is updated.

If you wish to clear the filtering criteria currently in use, click the **New Search** button whereby all filter criteria is removed.

3.4.2 The Table Area

The Table Area is used for listing configuration elements in order for you to locate and open the ones you need to work with. It is possible to scroll down an up through the lists.

The basic layout of the Table Area is the same for all tabs but the columns differ. Figure 3 below depicts the Table Area for the Risk Profiles tab where five risk profiles (configuration elements) are displayed as a result of a filtering operation.



Risk Profile ID	Keyword	User	Office	State	Activated At
3	Chapter 85	User9876	IEARK100	Active	2009-08-27 16:10
2	Garlic	User9876	IEARK100	Planned	
4	Mineralwateer from Nigeria	User9876	IEARK100	Active	2009-08-27 16:12
5	Mushroom from Eastern Europe	User9876	IEARK100	Active	2009-08-27 16:13
1	test	User9876	IEARK100	Expired	2009-08-27 09:07

Figure 3: Table Area for risk profiles

3.4.2.1 Selecting and Opening Configuration Elements

Configuration elements that need to be changed or updated can be opened from the Table Area. The method for selecting is the same in all tabs. When you have located such an element you select it in the list and then click the **Open** button, , on the Toolbar. Alternatively, instead of clicking the **Open** button you can select **Open** from the File menu or use the shortcut key combination CTRL-O. Another option is to double click on the element. The configuration element will open in the Data Area.

If you wish to open several configuration elements it is possible to first select them all and then open them using your preferred method for opening. The standard Windows option of selecting a block of elements and selecting several non-consecutive elements is also available.

If a block of elements is to be selected press the SHIFT key while clicking on the first and last element in the block. By pressing the CTRL key while clicking elements, non-consecutive elements can be highlighted as illustrated in Figure 4 below. All selected elements will open in the Data Area when you initiate the open command.

Country	Import	Export	Keyword
DK - Denmark	2 - Low	3 - Medium	EU info
BE - Belgium	2 - Low	3 - Medium	August Update
CY - Cyprus	4 - High	5 - Very High	August Update
ES - Spain	3 - Medium	1 - Very Low	July Update
FR - France	2 - Low	3 - Medium	August Update
GE - Germany	4 - High	2 - Low	May Update
GR - Greece	2 - Low	4 - High	June Update
IE - Ireland	1 - Very Low	1 - Very Low	July Update
LX - Luxemburg	3 - Medium	3 - Medium	EU info
ML - Malta	2 - Low	4 - High	EU info
NL - The Netherlands	5 - Very High	1 - Very Low	July Update
NO - Norway	2 - Low	3 - Medium	June Update
SE - Sweden	2 - Low	2 - Low	May Update
SF - Finland	1 - Very Low	1 - Very Low	June Update
UK - United Kingdom	2 - Low	4 - High	August Update
US - United States of America	5 - Very High	5 - Very High	July Update

Figure 4: Selecting block of elements and non-consecutive elements

When you have opened an element it is opened in read-only mode. If you wish to edit the an open element you then need to open it for edit (please note that this requires that you

have been assigned access right for edit) and in order to do this you click the **Edit** button, , on the Toolbar. Alternatively you can select the **Edit** menu item in the **Edit** menu or use the shortcut key combination CTRL-E.

3.4.2.2 Duplicating Configuration Elements

Instead of creating new configuration elements you can use the possibility of duplicating an existing element and change it according to your needs. All types of configuration elements can be duplicated though user rights may limit your access to do so.

To duplicate a configuration element you first select it and then click the **Duplicate** button, , on the Toolbar. Alternatively, instead of clicking the **Duplicate** button you can select **Duplicate** from the Edit menu or use the shortcut key combination CTRL-SHIFT-D. The duplicated configuration element will open in the Data Area.

Only single configuration elements can be duplicated – if you have selected more than one in the list only the one you clicked on last will be duplicated.

3.4.2.3 Deleting Configuration Elements

Configuration elements can, with one exception, be deleted. In the case of Risk Profiles only those that are in the Suspended state can be deleted.

To delete a configuration element you first select it and then click the **Delete** button, , on the Toolbar. Alternatively, instead of clicking the **Delete** button you can select **Delete** from the Edit menu or use the **Delete** key.

You can only delete one configuration element at a time; if you have selected several elements and try to delete them only the one with the lowest ID will be deleted.

3.4.2.4 Printing Configuration Elements

Configuration elements can be printed as lists or details for a single element. These printed reports each have a fixed format that is not user configurable. In order to print a list of configuration elements you first prepare the list by filtering the elements you wish to be included. You then have to select one line in the list in order to make the Table Area the active area.

If you wish to print the details of a configuration element in the Data Area you then must select it in the by clicking on the tab header of the element in Data Area.

In order to start the print process you click the **Print** button, , on the Toolbar. Alternatively you can select the **Print** menu item from the **Edit** menu or use the shortcut keyboard combination CTRL-P.

Before being printed on paper the printouts will be opened in a print preview window. In the ***Clearance Workbench*** the reports are created using HTML print and the preview will be presented in a HTML browser, see Figure 5 below. When you are finished reviewing the report in the browser you click the **Print** button to send the report to a printer.

For information about the functionalities and options available in the browser please refer to the user guide for the relevant browser.

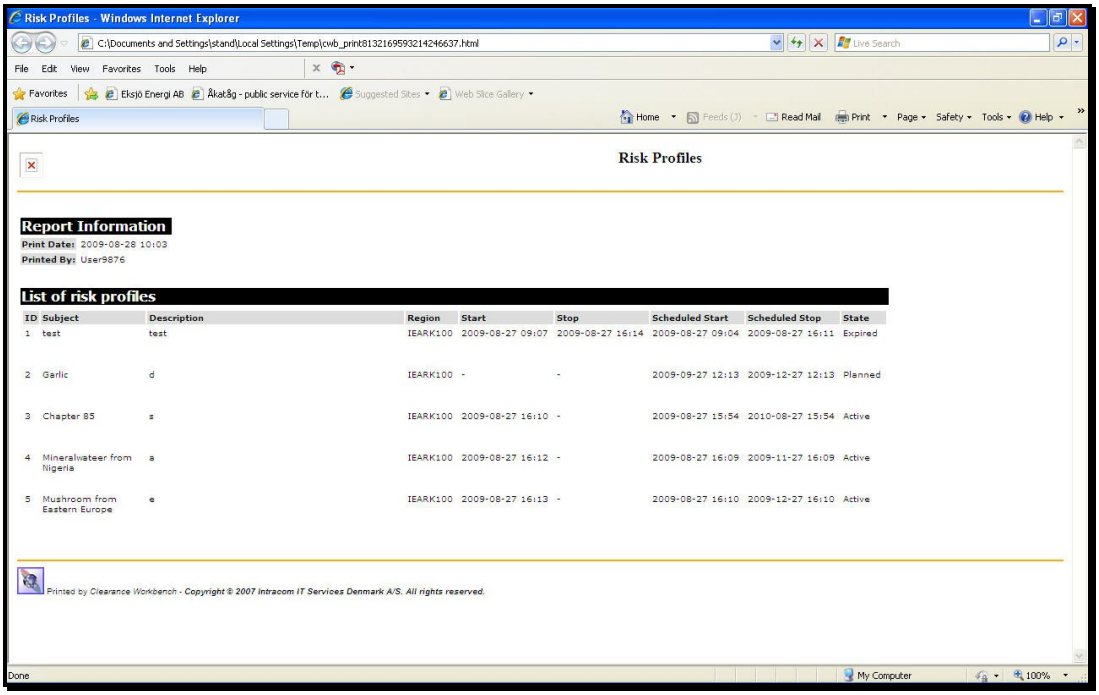

Risk Profiles - Windows Internet Explorer
Risk Profiles
Report Information
Print Date: 2009-08-28 10:03
Printed By: User9876
List of risk profiles
ID Subject Description Region Start Stop Scheduled Start Scheduled Stop State
1 test test IEARK100 2009-08-27 09:07 2009-08-27 16:14 2009-08-27 09:04 2009-08-27 16:11 Expired
2 Garlic d IEARK100 - - 2009-09-27 12:13 2009-12-27 12:13 Planned
3 Chapter 85 s IEARK100 2009-08-27 16:10 - 2009-08-27 15:54 2010-08-27 15:54 Active
4 Mineralwateer from Nigeria a IEARK100 2009-08-27 16:12 - 2009-08-27 16:09 2009-11-27 16:09 Active
5 Mushroom from Eastern Europe e IEARK100 2009-08-27 16:13 - 2009-08-27 16:10 2009-12-27 16:10 Active
Printed by Clearance Workbench - Copyright © 2007 Intracom IT Services Denmark A/S. All rights reserved.

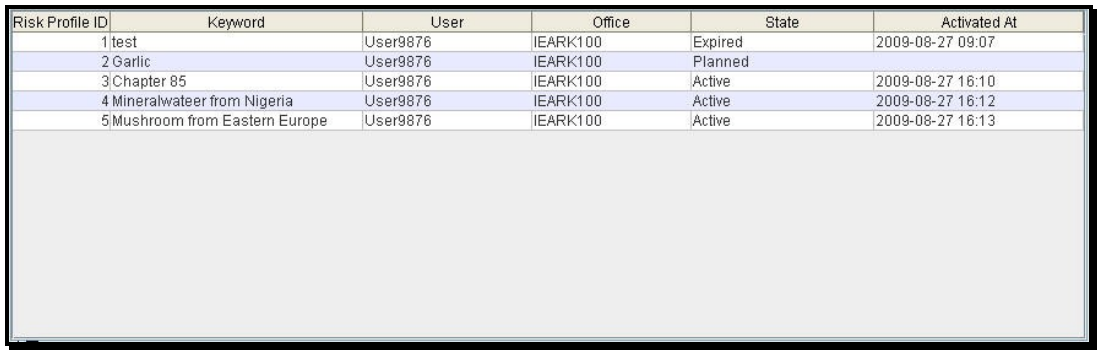
Figure 5: Preview of a report containing a list of Risk Profiles

☞ **Note:** Unfiltered lists in the Table Area can contain a considerable number of rows. Before launching a printout it is thus vital to limit the list by using the criteria in the Filter Area.

3.4.2.5 Moving and Changing Columns

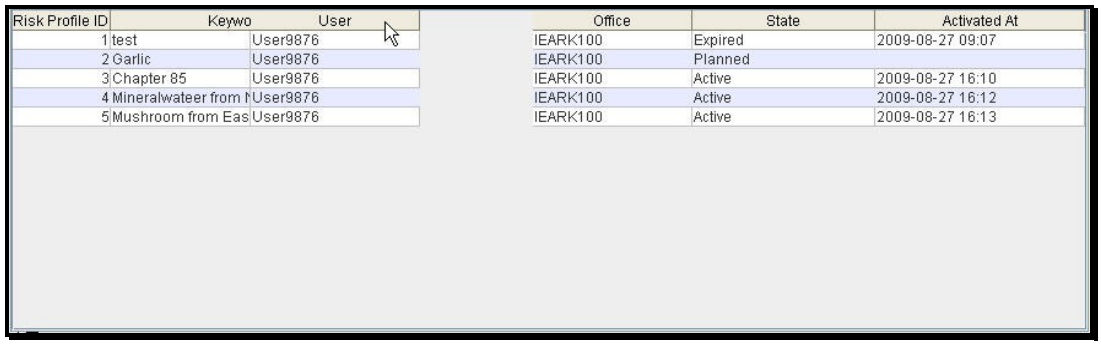
The columns viewed can be moved to suit your preference and facilitate your search. Columns are moved by placing the cursor on a column header, pressing the left mouse button and dragging the column to the desired position. As you move a column across another column the latter will jump to the opposite side of the column you are dragging. When you release the mouse button the column will then be placed in the position where the second column was placed before the moving operation. Figure 6 and Figure 7 below show such a moving operation. The User column is dragged from a position to the right of the Keyword column and the Keyword column has in Figure 7 just jumped to the right of the User column.

☞ **Tip:** It is not necessary to drag a column all the way and adjust it in the vacant slot that the jumping column has created. As soon as the other column has jumped into position you can release the mouse button and the dragged column will automatically be adjusted into the vacant slot.



Risk Profile ID	Keyword	User	Office	State	Activated At
1	test	User9876	IEARK100	Expired	2009-08-27 09:07
2	Garlic	User9876	IEARK100	Planned	
3	Chapter 85	User9876	IEARK100	Active	2009-08-27 16:10
4	Mineralwateer from Nigeria	User9876	IEARK100	Active	2009-08-27 16:12
5	Mushroom from Eastern Europe	User9876	IEARK100	Active	2009-08-27 16:13

Figure 6: Illustrates the columns before moving

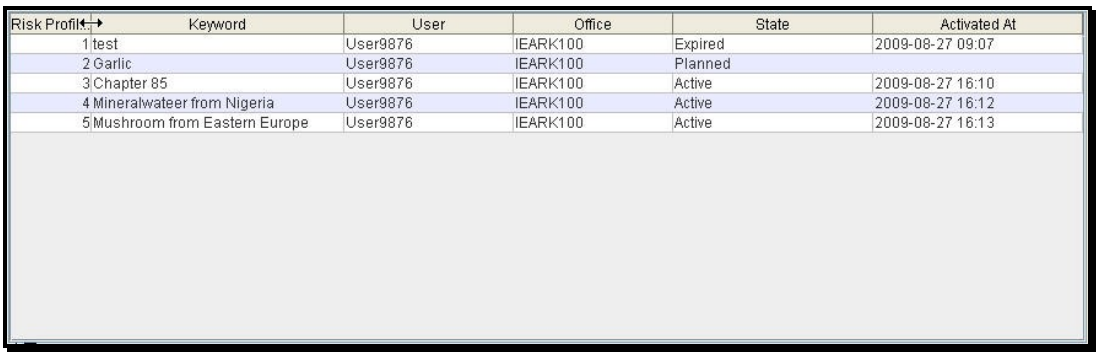


Risk Profile ID	Keywo	User	Office	State	Activated At
1	test	User9876	IEARK100	Expired	2009-08-27 09:07
2	Garlic	User9876	IEARK100	Planned	
3	Chapter 85	User9876	IEARK100	Active	2009-08-27 16:10
4	Mineralwateer from N	User9876	IEARK100	Active	2009-08-27 16:12
5	Mushroom from Eas	User9876	IEARK100	Active	2009-08-27 16:13

Figure 7: Illustrates the column move in progress

The column width can be altered by moving the cursor across the lines dividing the column headers. The cursor will then change shape to a double-pointed arrow, .

The column width can then be adjusted by pressing the left mouse button and dragging the cursor in the desired direction. In Figure 8 below the width of the Risk Profile ID column is in the process of being changed.



Risk Profile	Keyword	User	Office	State	Activated At
1	test	User9876	IEARK100	Expired	2009-08-27 09:07
2	Garlic	User9876	IEARK100	Planned	
3	Chapter 85	User9876	IEARK100	Active	2009-08-27 16:10
4	Mineralwateer from Nigeria	User9876	IEARK100	Active	2009-08-27 16:12
5	Mushroom from Eastern Europe	User9876	IEARK100	Active	2009-08-27 16:13

Figure 8: Illustrates changing the column width

3.4.2.6 Sorting Table Area Contents

The contents of the Table Area, i.e. the configuration elements, can be sorted to facilitate your work in finding specific elements. It is possible to sort ascending or descending and you can use sublevels to execute multiple column sorting. The sorting is performed the same way for all tabs.

To sort the configuration elements in a Table Area in ascending order for a certain column you click on the header for the column to be sorted.

Depending on whether the column was sorted or not before clicking the header, the result will be the following:

- If the column was not sorted before clicking, the elements will be sorted in ascending sort order (A-Z) - another click will resort in descending sort order (Z-A).
- If the column was sorted before clicking, the records will be resorted in the opposite sort order.

When you have clicked on a header to invoke sorting a blue arrow will appear in the header to indicate that the column is sorted and also whether the sorting is ascending or descending. If the arrow is pointing upwards the column is sorted ascending and if it is pointing downwards it is sorted descending. Figure 9 below illustrates the Table Area contents for Country Code Risk Objects sorted ascending using the risk values for import as key for the sorting.

Country	Import	Export	Keyword
IE - Ireland	1 - Very Low	1 - Very Low	July Update
SF - Finland	1 - Very Low	1 - Very Low	June Update
UK - United Kingdom	1 - Very Low	1 - Very Low	July Update
DK - Denmark	2 - Low	3 - Medium	EU info
BE - Belgium	2 - Low	3 - Medium	August Update
FR - France	2 - Low	3 - Medium	August Update
GR - Greece	2 - Low	4 - High	June Update
ML - Malta	2 - Low	4 - High	EU info
NO - Norway	2 - Low	3 - Medium	June Update
ES - Spain	3 - Medium	1 - Very Low	July Update
LX - Luxemburg	3 - Medium	3 - Medium	EU info
CY - Cyprus	4 - High	5 - Very High	August Update
GE - Germany	4 - High	2 - Low	May Update
NL - The Netherlands	5 - Very High	1 - Very Low	July Update
US - United States of America	5 - Very High	5 - Very High	July Update

Figure 9: Illustration of the Table Area contents for Country Code Risk Objects sorted ascending

In order to use sublevels for sorting you first have to indicate the primary column for sorting by clicking on the header, then press and hold down the SHIFT key and click the header for the first column to be used as the first sublevel for sorting. The result is that this column will be sorted ascending for those configuration elements in the primary column that occurs more than once. By clicking on the header again the sort order for the sublevel will be changed to descending. The result of this operation is illustrated in Figure 10 below. In this example the primary column for sorting is the Import column and the sort order is ascending. The Country code column is used as sublevel for sorting and the order is descending.

Country	Import	Export	Keyword
UK - United Kingdom	1 - Very Low	1 - Very Low	July Update
SF - Finland	1 - Very Low	1 - Very Low	June Update
IE - Ireland	1 - Very Low	1 - Very Low	July Update
NO - Norway	2 - Low	3 - Medium	June Update
ML - Malta	2 - Low	4 - High	EU info
GR - Greece	2 - Low	4 - High	June Update
FR - France	2 - Low	3 - Medium	August Update
DK - Denmark	2 - Low	3 - Medium	EU info
BE - Belgium	2 - Low	3 - Medium	August Update
LX - Luxemburg	3 - Medium	3 - Medium	EU info
ES - Spain	3 - Medium	1 - Very Low	July Update
GE - Germany	4 - High	2 - Low	May Update
CY - Cyprus	4 - High	5 - Very High	August Update
US - United States of America	5 - Very High	5 - Very High	July Update
NL - The Netherlands	5 - Very High	1 - Very Low	July Update

Figure 10: Illustration of sorting using sublevels

If you wish to remove columns from the sorting you press and hold down CTRL and click the header for the column to be revoked from sorting.

The number of columns to be used for sublevel sorting is not limited; all available columns can be used.

3.4.2.7 Hiding and Displaying Columns

The columns in the Table Areas can be hidden and displayed to facilitate your current needs at any time. In order to hide or display a column you right click on any of the headers in the Table Area and select the **Change visible columns** menu item, see below**.**

As a result the **Change Visible Columns** dialog is opened and here you can tick and un-tick the tick boxes in order to hide and display the desired columns. For tick boxes that are un-ticked the matching columns will be hidden and for tick boxes that you have ticked those columns will be displayed. When ready click the **OK** button to close the dialog. If you wish to abort the operation click the **Cancel** button. Figure 11 below depicts the **Change Visible Columns** dialog for the Risk Profiles tab.

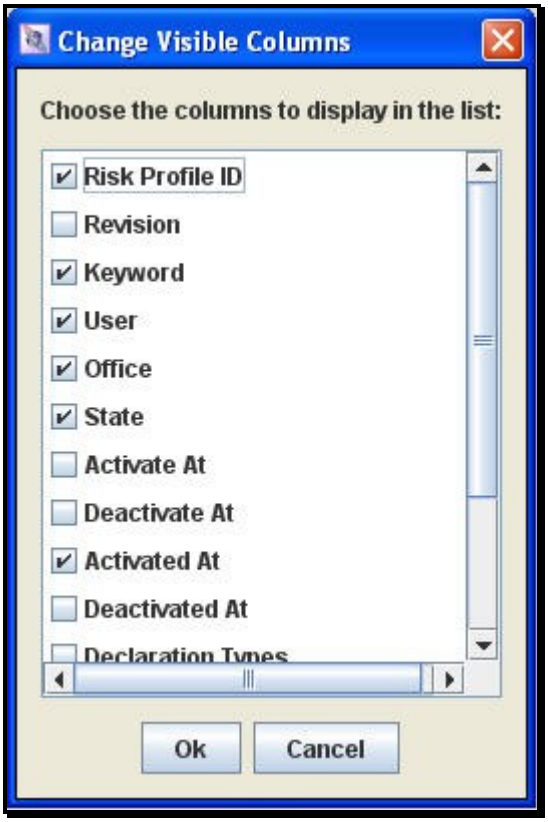


*Figure 11: The **Change Visible Columns** dialog for the Risk Profiles tab*

3.4.3 The Data Area

In each of the main tabs the Data Area is where the work of defining, changing and updating of the configuration elements is executed. The contents of the Data Area are to a very large extent tab specific but there is some generic functionalities that are described below.

A configuration element can be displayed in the Data Area in two different modes; opened as read-only or opened for edit. An element in read-only mode is not possible to change in

any way; the user can only view the contents of the element. An element opened for edit can however be changed by the user and the extent of possible changes varies from Main tab to Main tab.

Depending on user rights a user can be given access to either read-only mode or both read-only and edit modes.

3.4.3.1 Displaying Configuration Elements

In order to display an existing configuration element in the Data Area you select it from the Table Area and then click the **Open** button, , on the Toolbar. Alternatively, instead of clicking the **Open** button you can select **Open** from the File menu or use the shortcut key combination CTRL-O. Another option is to double click on the element.

If you wish to open several configuration elements it is possible to first select them all and then open them using your preferred method for opening. The standard Windows option of selecting a block of elements and selecting several non-consecutive elements is also available.

If a block of elements is to be selected press the SHIFT key while clicking on the first and last element in the block. By pressing the CTRL key while clicking elements, non-consecutive elements can be highlighted. All selected elements will open in the Data Area when you initiate the open command. The number of elements that can be displayed in the Data Area is not limited.

In the Data Area configuration elements are opened in separate tabs and if you display more elements than can fit into the Data Area, scroll buttons will appear and these can be used to scroll back and forth among the elements.

☞ **Note:** If the result of a filtering operation is just one configuration element, which is listed in the Table Area, then the element will automatically be displayed in the Data Area.

When configuration elements are opened in the Data Area they are opened in read-only mode. If you wish to edit them click the **Edit** button, , on the Toolbar. Alternatively, instead of clicking the **Edit** button you can select **Edit** from the Edit menu or use the shortcut key combination CTRL-E.

New configuration elements will be displayed in the Data Area when you have clicked the **New** button, , on the Toolbar. Alternatively, instead of clicking the **New** button you can select **New** from the File menu or use the shortcut key combination CTRL-N.

3.4.3.2 Closing Configuration Elements

To close configuration elements you click the **Close** button, , in the top right hand corner of the tab.

4. General Configuration Items

The configuration work performed in the ***Clearance Workbench*** consists mainly of working with configuration items such as text boxes, tick boxes and drop down boxes. This work is carried out in the Filter Area and the Data Area and in these areas there are both general configuration items and specific configuration items. Typically general configuration items are to be found in more than one tab (Main tabs and sub-tabs) and working with these are the same irrespective of what tab you have selected. Specific configuration items however are found only in one tab. This section describes the general configuration items and how you work with these. The specific configuration items are described in the sections covering the tabs where they appear.

4.1 Declaration Types

In some of the main tabs it is possible to select one or more declaration types. This can be done in three different ways, either by use of tick boxes in a list, tick boxes in a tree-structure or selection from drop down boxes. Tick boxes allow for the selection of more than one declaration type while using drop down boxes limits the selection to only one declaration type.

Figure 12 below illustrates tick boxes in a list for selecting declaration types. In the figure types B, D and I are selected. Clicking a selected check box de-selects the item and removes the tick.

Export: (A) Export / Dispatch
Export: (B) Customs warehousing
Export: (C) Re-export after economic procedure
Export: (D) Re-export after warehousing
Export: (E) Outward processing
Import: (H) Release for free circulation
Import: (I) Customs procedure - economic impact
Manifest Declarations
Warehousing: (J) Customs warehousing type A, B, C, E, F
Warehousing: (K) Customs warehousing - other

Figure 12: Selection of declaration types from tick boxes

Selecting declaration types using tick boxes in a tree-structure is shown in Figure 13 below. By default these structures are expanded when you start the ***Clearance Workbench.*** For convenience you can collapse the structure by clicking the **Collapse** button, , and expand them with the **expand** button, . When you have opened the relevant parts of the structure you then click the tick boxes to select declaration segments.

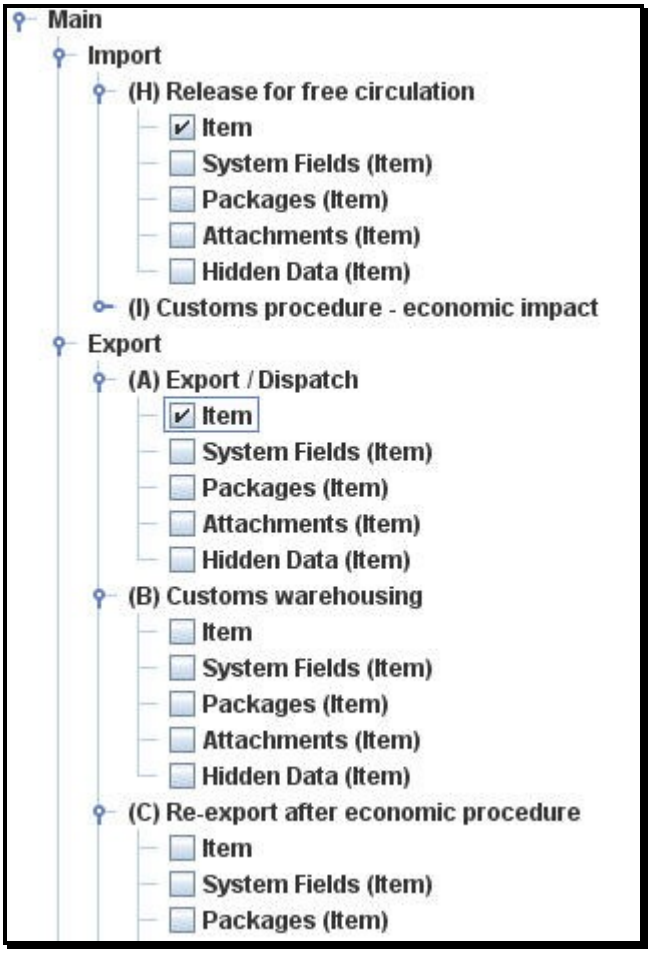


Figure 13: Selection of declaration types from tick boxes in a tree-structure

In Figure 14 below the selection of declaration type is made via a drop down box. In order to make the selection the drop down box must be opened by clicking anywhere in the box (the top line). The selection is made by clicking on the relevant declaration type. This will close the drop down list.
Note the scroll bar on the right side of the drop down list.

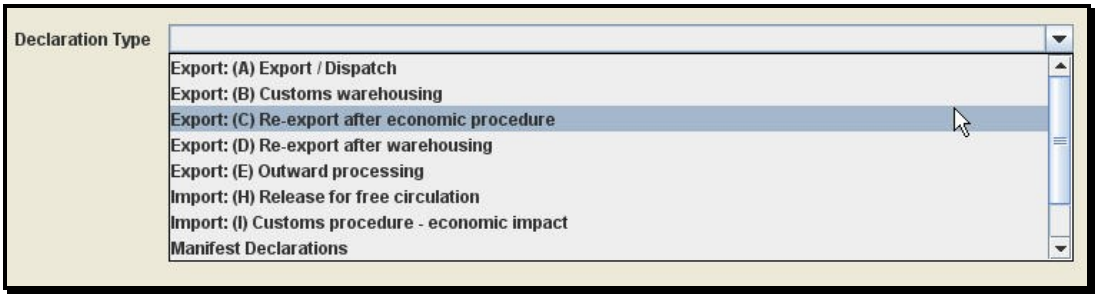

Declaration Type
Export: (A) Export / Dispatch
Export: (B) Customs warehousing
Export: (C) Re-export after economic procedure
Export: (D) Re-export after warehousing
Export: (E) Outward processing
Import: (H) Release for free circulation
Import: (I) Customs procedure - economic impact
Manifest Declarations

Figure 14: Selection of declaration types from drop down box

4.2 Declaration Segments

The declarations that are risk analyzed in the ESKORT Clearance Risk Analysis system are divided into segments, e.g. the Item segment, the Attachments segment and the System segment. In the ***Clearance Workbench*** the declaration segments are presented in declaration types outlines with a structure that holds the four main declaration types – Import, Export, Warehousing and Manifest. Within each main type the declaration segments are presented as tick boxes.

Figure 15 illustrates the declaration types outline where some tick boxes for selecting declaration segments are ticked. The selection is made by clicking the relevant declaration segment tick boxes. Clicking once again removes the tick and the declaration segment is thus de-selected.

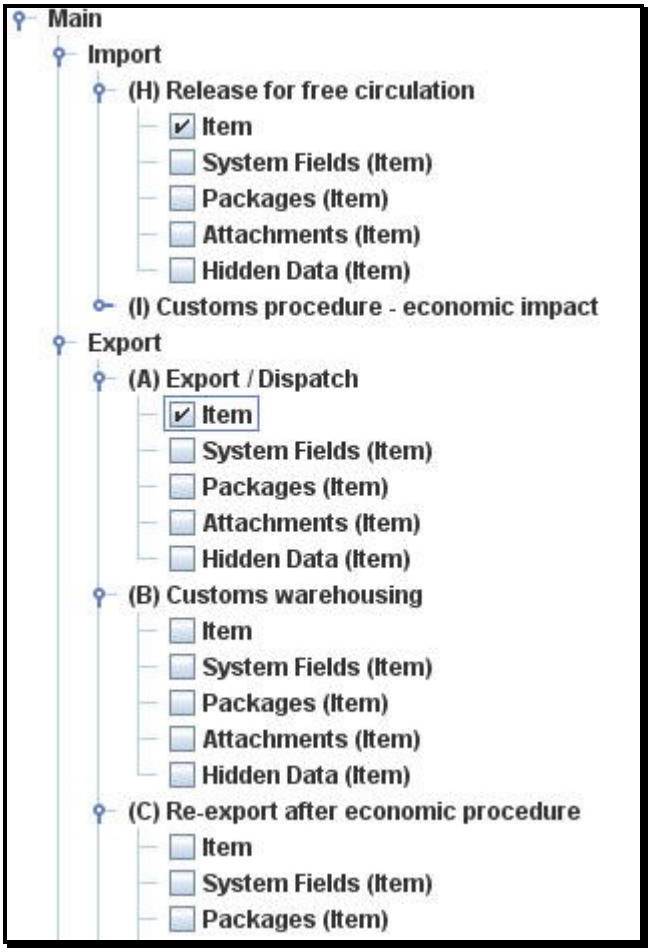


Figure 15: Selection of declaration segments from tick boxes

4.3 Date and Time

In some work such as defining a new risk profile, date and time must be entered and this is carried out in a **Date Editor**. Figure 16 illustrates the **Date Editor** dialog which can be used to enter dates.

Figure 16: Date Editor for defining date and time

In order to define a time you select month and year from the dropdown boxes. You then indicate the day by clicking on the relevant day. The time can be selected from the drop down box, you can enter the time directly into the top line in the drop down box or you can click the **Now** button. When you have finished you click the **OK** button.

4.4 Customs Office

Selection of customs office is done by ticking the **Customs Office** tick box and entering the relevant code for the customs office in the text box. It is possible to use wildcards in the text box to make easier to indicate a customs office for which you do not know the full code. For further instructions refer to section 4.9 Wildcards.

☞ **Note:** Instead of ticking the tick boxes you can click in the text box and as a result the tick box is ticked and the text box is opened for edit.

4.5 Keyword

Keywords are used for identifying configuration elements in order to make it easier to locate elements stored in the system. Each element also has an identification number that can be used but it is generally easier to locate elements with the help of keywords, especially if a uniform naming convention is applied.

Assigning keyword is done by ticking the **Keyword** tick box whereby the **Keyword** text box is opened and you can then enter the keyword name. It is possible to use wildcards in the text box to make easier to indicate keyword for which you do not know the full name. For further instructions refer to section 4.9 Wildcards.

☞ **Note:** Instead of ticking the tick boxes you can click in the text box and as a result the tick box is ticked and the text box is opened for edit.

4.6 Trader Number

Indication of trader number is done by ticking the **Trader** tick box whereby the **Trader** text box is opened and you can then enter the trader number. It is possible to use wildcards in the text box to make easier to indicate traders for which you do not know the full number. For further instructions refer to section 4.9 Wildcards.

☞ **Tip:** Instead of ticking the tick boxes you can click in the text box and as a result the tick box is ticked and the text box is opened for edit.

4.7 User

Indication of user is done by ticking the **User** tick box whereby the **User** text box is opened and you can then enter the user ID. It is possible to use wildcards in the text box to make easier to indicate users for which you do not know the full ID. For further instructions refer to section 4.9 Wildcards.

☞ **Tip:** Instead of ticking the tick boxes you can click in the text box and as a result the tick box is ticked and the text box is opened for edit.

4.8 Hyperlinks

In the ***Clearance Workbench*** there are a number of text boxes for entering large amounts of text such as descriptions or remarks. It is possible to use hyperlinks in these boxes for the purpose of referring to specific information. For working with hyperlinks you have the following options:

- Insert a hyperlink
- Edit a hyperlink
- Open a hyperlink

To get access to these options you right click inside the relevant text box and as a result the following menu opens:

Figure 17: The hyperlink menu

4.8.1 Create a Hyperlink

In order to create a hyperlink you right click anywhere inside the text box where you wish to place the hyperlink and select **Insert hyperlink. A**s a result the **Edit hyperlink** dialog is opened. In the **URL** text box you enter the address to the URL you wish to link to and optionally a title in the **Title** text box. Click the **OK** button and the title will then be displayed as the hyperlink. If you select to omit the title in the **Edit hyperlink** dialog, the URL will be displayed as hyperlink. Figure 18: Creating a hyperlink below shows the **Edit hyperlink** dialog with URL and title.

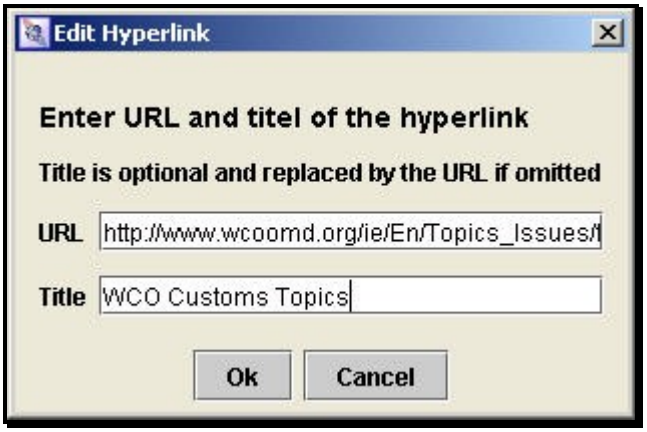


Figure 18: Creating a hyperlink

4.8.2 Edit a Hyperlink

In order to edit a hyperlink you right click on the hyperlink and select **Edit hyperlink. A**s a result the **Edit hyperlink** dialog is opened displaying the properties of the hyperlink. Edit the hyperlink and click the **OK** button to save your work.

4.8.3 Open a hyperlink

To open the hyperlink, right click on it and select **Open hyperlink** as illustrated in Figure 19 below.

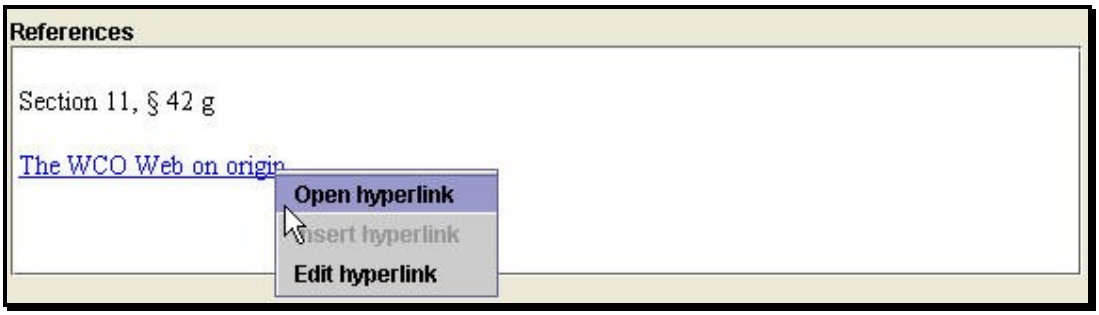


Figure 19: Opening a hyperlink

4.9 Wildcards

It is possible to use wildcards in certain parts of the ***Clearance*** *Workbench* such as e.g. text boxes for filtering.. The wildcards available for use are the asterisk character (*) or the question mark character (?). The question mark substitutes one character in the exact position of the question mark. The asterisk substitutes one or more characters in front of or after the position of the asterisk.

The Figure 20 and Figure 21 below illustrate using the ? and * wildcards to search for customs office IEDUB999:

Figure 20: Illustration of using the asterisk wildcard

Figure 21: Illustration of using the question mark wildcard

5. Working with Risk Profiles

In Phase 1 of the risk analysis process risk profiles are used for trapping specific shipments that matches conditions defined in the Risk Profiles configuration elements. Risk Profile conditions can be created by combining declaration fields with operators indicating values. In the ***Clearance Workbench*** risk profiles are defined and maintained in the Risk Profiles Tab, refer to Figure 22 below.

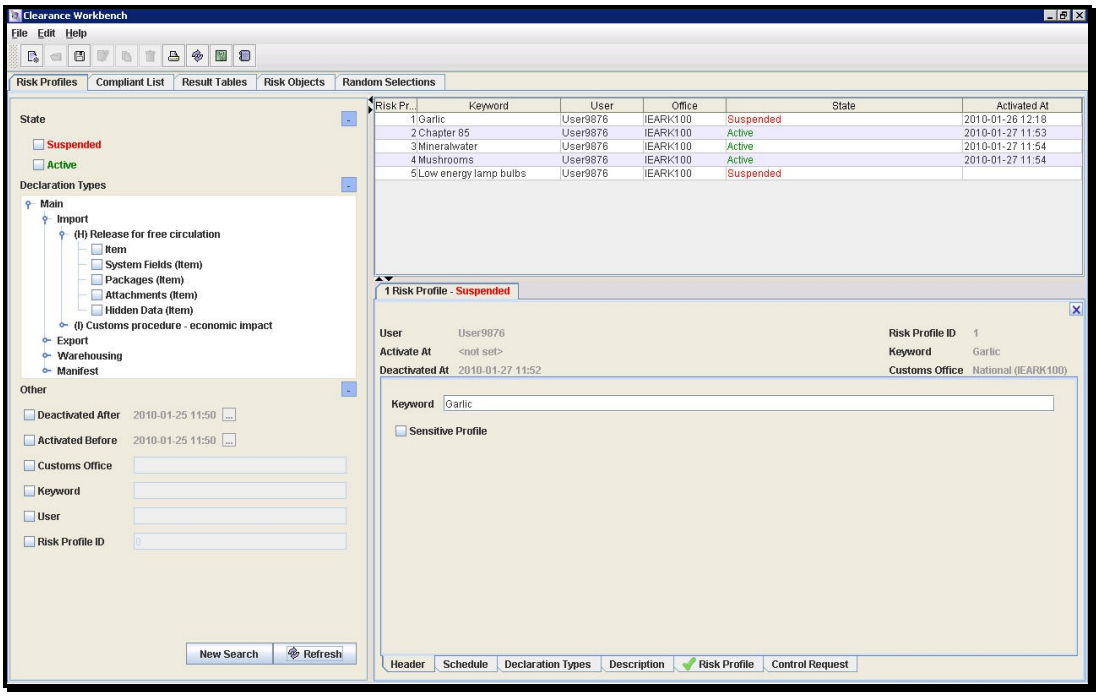


Figure 22: The Risk Profiles tab

The details pertaining to a risk profile are displayed in one of six sub-tabs in the Data Area; the Header sub-tab, the Schedule sub-tab, the Declaration Types sub-tab, the Description sub-tab, the Risk Profile sub-tab and the Control Request sub-tab.

Risk Profiles can exist in two different states, the active state or the suspended state. When you define a new risk profile you have to set one of these two states and when you save the risk profile it is then moved into that state. You can at any later point in time change the state. All risk profiles in the active state are used by the ***Risk Analysis Server*** in the analysis process as opposed to those in the suspended state which are not used. It is also possible to schedule risk profiles so that they are automatically activated and suspended according to predefined patterns, e.g. be active two hours every Monday starting at 09:00 AM.

Basically a risk profile can be applied nationwide meaning that all declarations passing through all customs offices will be tested against the risk profile or it can be applied to specific offices. Risk profiles that are defined by officers in the national customs office will by default apply nationwide. You can however limit a risk profile to a specific customs office or selected offices by using the customs office code/codes as a condition in the risk profile. Risk profiles defined by officers in local customs offices will apply to declarations filed at the local offices only.

5.1 Define a new Risk Profile

To define a new risk profile you click the **New** button, , on the Toolbar. Alternatively, instead of clicking the **New** button you can select **New** from the File menu or use the shortcut key combination CTRL-N. As a result a new, empty risk profile is opened as a tab in the Data Area. The sub-tabs at the bottom of the screen are used for entering the details about the risk profile and you can work with these sub-tabs in any order with the exception of the Risk Profile sub-tab.. In order to be able to work in the Risk Profile sub-tab you first have to select a declaration type in the Declaration sub-tab. When you create a new risk profile the Header sub-tab is by default the active sub-tab.

5.1.1 The Header sub-tab

Figure 23 below illustrates the Header sub-tab for a new risk profile.

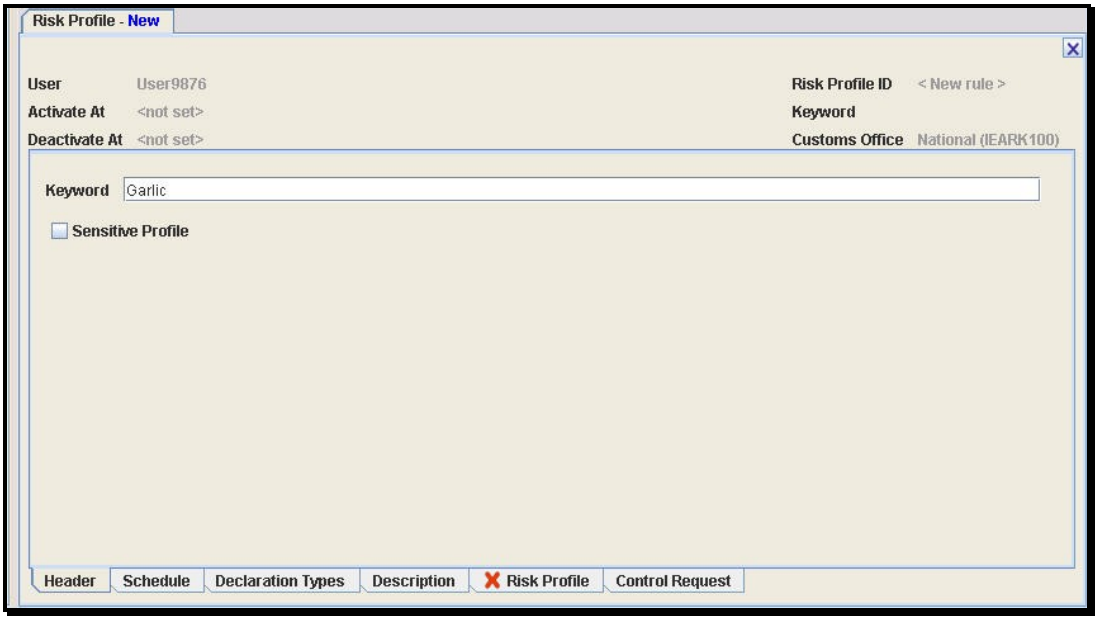


Figure 23: The Header sub-tab for a new risk profile

The top part of the sub-tab contains basic information about the risk profile and when a new risk profile is created the user information is automatically set to the logon user name. The other information parts will be filled out when the risk profile has been saved.

Below the basic information there is a text box for setting keyword and a tick box for indicating sensitive profile status. It is mandatory to enter a keyword but using the tick box is optional.

The **Keyword** text box is used for entering a keyword that can be used as an alphabetic identification of the risk profile. Refer to section 4.5 for instructions about keywords.

The **Sensitive Profile** tick box can be used for restricting access to the risk profile you are defining to users in the national role group only. Users belonging to the local role group will not be able to see this risk profile, not even in the Table Area.

5.1.2 The Schedule sub-tab

The Schedule sub-tab is used for scheduling the risk profile, i.e. when the risk profile should be active or suspended. Figure 24 below illustrates the Schedule sub-tab for a new risk profile.

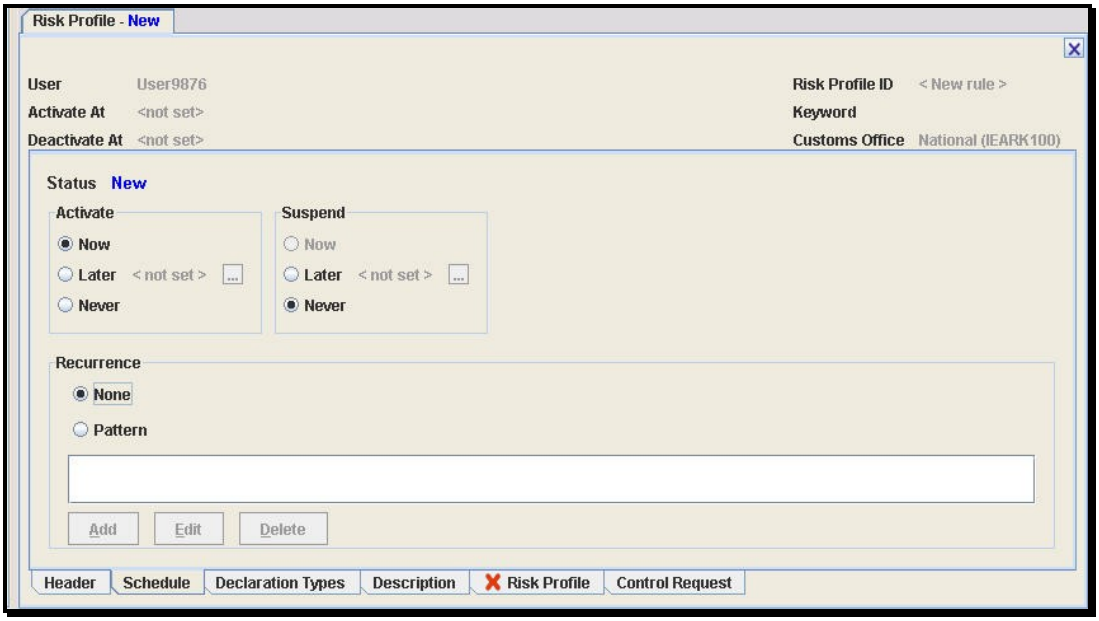


Figure 24: The Schedule sub-tab

5.1.2.1 General Activation and Suspension

The **Activate** and **Suspend** boxes are used for setting the times for activation and suspension of the risk profile respectively.

Depending on the state of the risk profile the radio buttons in the boxes are enabled or disabled. This is described in the subsequent three sections.

5.1.2.1.1 Scheduling a new Risk Profile

A new risk profile can be scheduled to become active:

- Immediately (as soon as the new risk profile is saved)
- Later (when a specific time occurs)
- Unspecified (the new risk profile remains suspended until further action is taken by the user)

Immediately Activation

A risk profile can be scheduled for immediately activation. This is done by selecting the **Now** button in the **Activate** box. In that case the **Suspend** box provides the options of choosing when the risk profile shall be suspended; this can be: **Later** or **Never**.

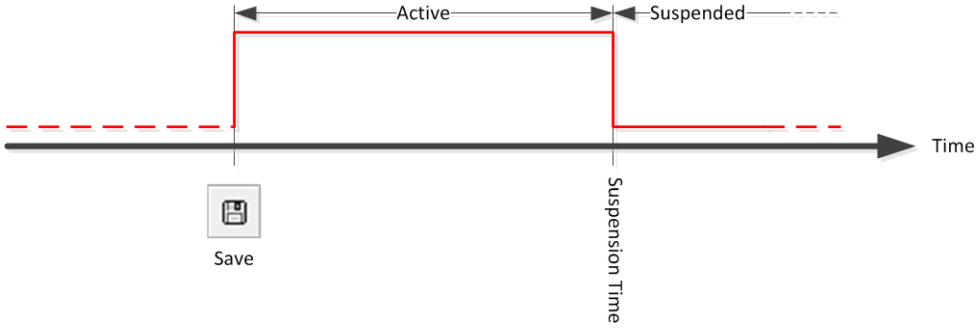


Figure 25: A new risk profile being immediately activated and scheduled for later suspension.

If **Later** is selected a specific date and time must be selected by clicking the button right to the **Later** button. After having saved the risk profile it will become immediately active and suspend automatically (**Suspend Later**) when the date and time selected by the calendar right to the **Later** button occurs (see Figure 25).

Figure 26: A new risk profile being immediately activated without a schedule for suspension.

If **Never** is selected in the **Suspend** box the risk profile will remain active (see Figure 26) until a user later reconfigures the risk profile to suspend it manually.

Later Activation

A risk profile can be scheduled for later activation. This is done by selecting the **Later** button in the **Activate** box and then select a specific date and time by clicking the button right to the **Later** button. In this case the **Suspend** box provides the option of choosing when the risk profile shall be suspended; this can be: **Later** or **Never**.

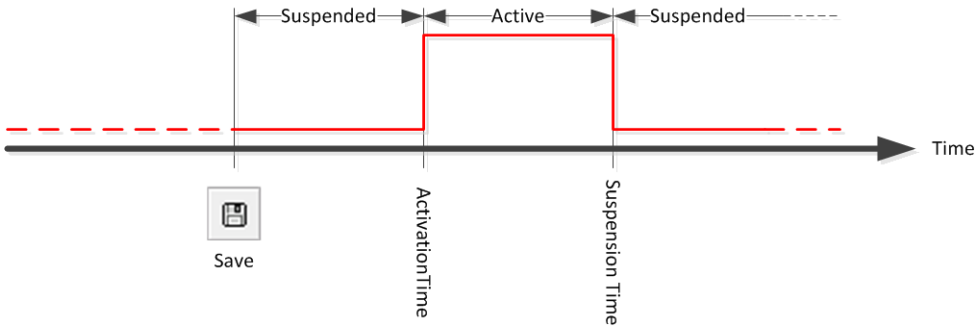


Figure 27: A new risk profile scheduled for later activation and later suspension.

If **Later** is selected a specific date and time must be selected by clicking the button right to the **Later** button in the **Suspend** box. After having saved the risk profile it will become active when the when the date and time selected by the calendar right to the **Later** button in the **Activate** box occurs and suspend automatically (**Suspend Later**) when the date and time selected by the calendar right to the **Later** button in the **Suspend** box occurs (see Figure 27).

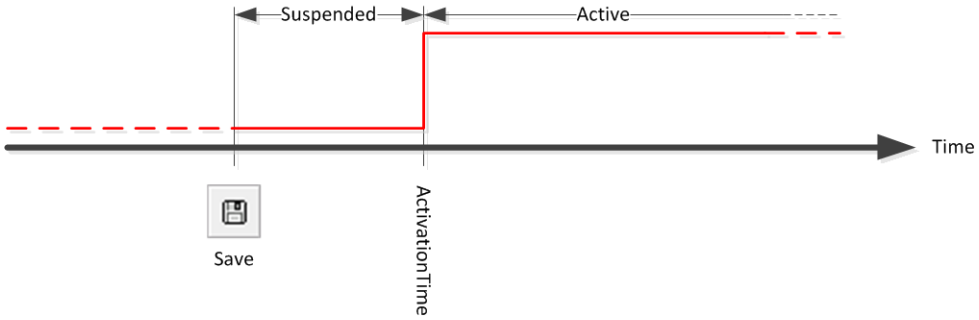


Figure 28: A new risk profile scheduled for later activation but without any specific suspension time.

If **Never** is selected in the **Suspend** box the risk profile will remain active until a user later reconfigures the risk profile to suspend it manually (see Figure 28).

Unspecified Activation

A risk profile can be created without any specified activation date. This is done by selecting the **Never** button in the **Activate** box. In that case the **Suspend** box will not provide any options.

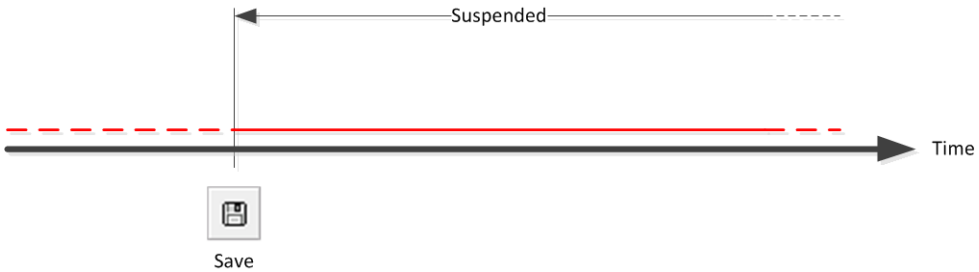


Figure 29: A new risk profile not yet scheduled to be active.

When saving the risk profile it will remain suspended until a user decides to activate it manually (see Figure 29).

5.1.2.1.2 Suspending an Active Risk Profile

When a risk profile is active it can be suspended by the user in one of the following ways:

- Immediately (as soon as the risk profile is saved)
- Later (when a specific time occurs)
- Unspecified (the risk profile remains active until further action is taken by the user)

Immediately Suspension

An active risk profile can be scheduled for immediately suspension. This is done by selecting the **Now** button in the S**uspend** box. In that case the **Activate** box provides the options of choosing when the risk profile shall be reactivated; this can be: **Later** or **Never**.

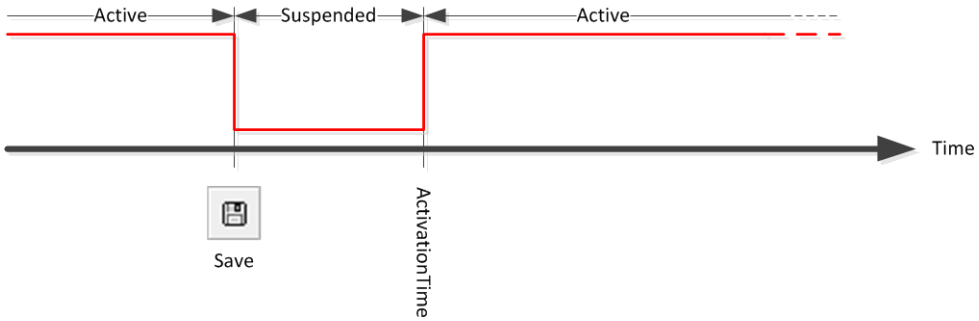


Figure 30: An active risk profile being immediately suspended but scheduled for later reactivation.

If **Later** is selected a specific date and time must be selected by clicking the ... button right to the **Later** button (see also section 5.1.2.1.4). When saving the risk profile it will suspend immediately and reactivate automatically (**Activate Later**) when the date and time selected by the calendar right to the **Later** button occurs (see Figure 30).

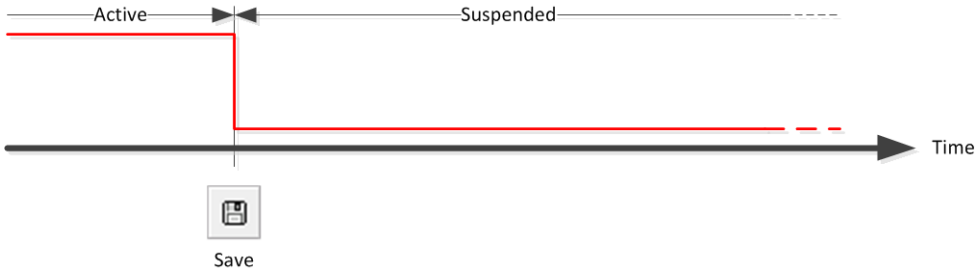


Figure 31: An active risk profile being immediately suspended with no scheduled reactivation.

If **Never** is selected in the **Activate** box the risk profile will remain suspended (**Activate Never**) until a user decides to activate it manually (see Figure 31).

Later Suspension

An active risk profile can be scheduled for later suspension. This is done by selecting the **Later** button in the **Suspend** box and then select a specific date and time by clicking the ... button right to the **Later** button (see also section 5.1.2.1.4). In that case the **Activate** box provides no further options before the selected date and time of the suspension occurs.

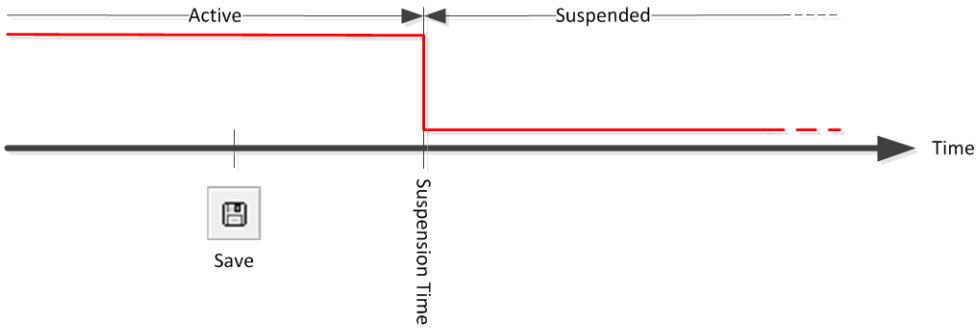


Figure 32: An active risk profile being scheduled for later suspension.

Unspecified Suspension

An active risk profile can remain active without having a specific schedule for when it shall be suspended. This is done by selecting the **Never** button in the **Suspend** box. In that case the **Activate** box will not provide any options.

When saving the risk profile it will remain active until a user decides to suspend it manually (see Figure 33).

Figure 33: An active risk profile without any scheduled suspension.

5.1.2.1.3 Reactivating a Suspended Risk Profile

When a Risk Profile is suspended it can be reactivated by the user in one of the following ways:

- Immediately (as soon as the risk profile is saved)
- Later (when a specific time occurs)
- Unspecified (the risk profile remains suspended until further action is taken by the user)

Immediately Reactivation

A suspended risk profile can be scheduled for immediately reactivation. This is done by selecting the **Now** button in the **Activate** box. In that case the **Suspend** box provides the options of choosing when the risk profile shall be re-suspended; this can be: **Later** or **Never**.

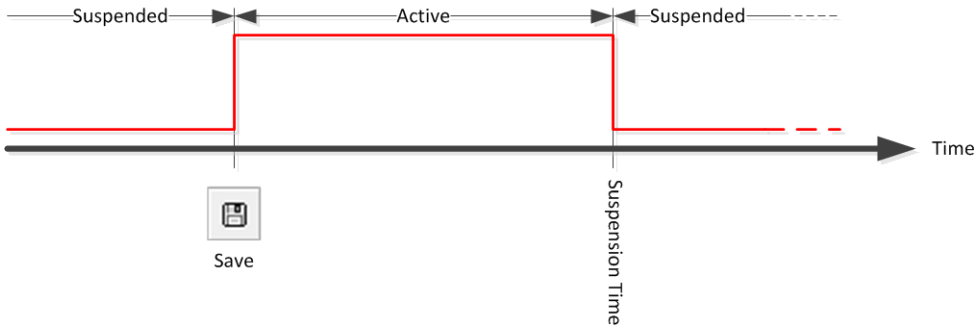


Figure 34: A suspended risk profile being immediately activated with a scheduled suspension time.

If **Later** is selected a specific date and time must be selected by clicking the ... button (see also section 5.1.2.1.4) right to the **Later** button. When saving the risk profile it will activated immediately and re-suspended automatically (**Suspend Later**) when the date and time selected by the calendar right to the **Later** button occurs (see Figure 34).

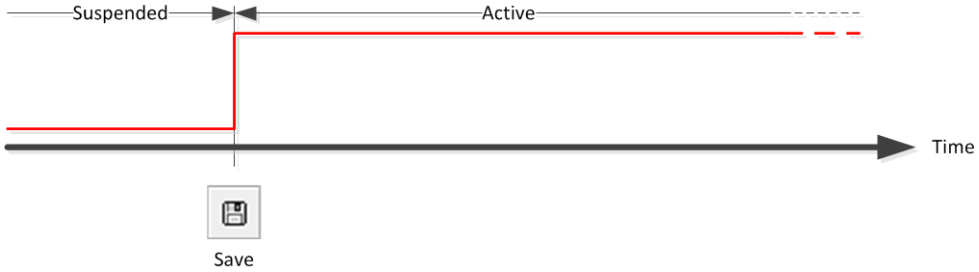


Figure 35: A suspended risk profile being immediately activated but without any scheduled suspension time.

If **Never** is selected in the **Suspend** box or it will remain active (**Suspend Never**) until a user decides to suspend it manually (see Figure 35).

Later Reactivation

A risk profile can be scheduled for later reactivation. This is done by selecting the **Later** button in the **Activate** box and then select a specific date and time by clicking the ... button (see also section 5.1.2.1.4) right to the **Later** button. In that case the **Suspend** box provides the options of choosing when the risk profile shall be re-suspended; this can be: **Later** or **Never**.

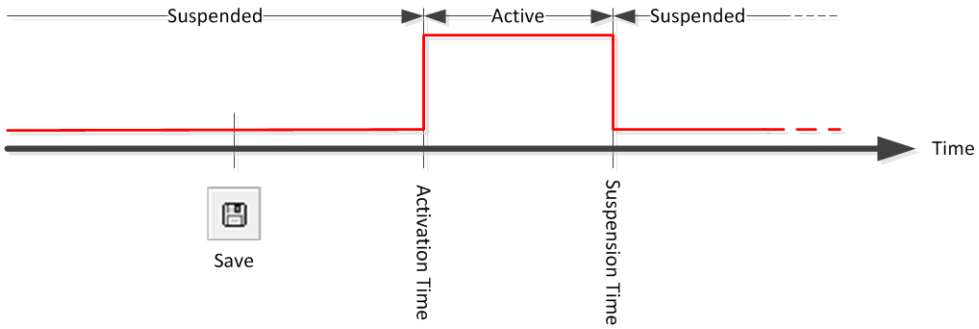


Figure 36: A suspended risk profile being planned for later activation and suspension.

If **Later** is selected a specific date and time must be provided by clicking the ... button (see also section 5.1.2.1.4) right to the **Later** button in the **Suspend** box. When saving the risk profile it will become active when the when the date and time selected by the calendar right to the **Later** button in the **Activate** box occurs and suspend again automatically (**Suspend Later**) when the date and time selected by the calendar right to the **Later** button in the **Suspend** box occurs (see Figure 36).

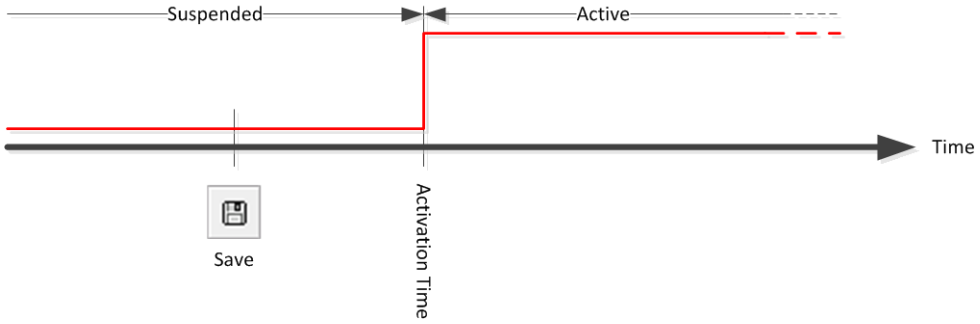


Figure 37: A suspended risk profile being scheduled for later activation but without any scheduled suspension.

If **Never** is selected in the **Suspend** box the risk profile will remain active until a user decides to suspend it manually (see Figure 37).

Unspecified Reactivation

A risk profile can be held suspended without any specified date for reactivation. This is done by selecting the **Never** button in the **Activate** box. In that case the **Suspend** box will not provide any options.

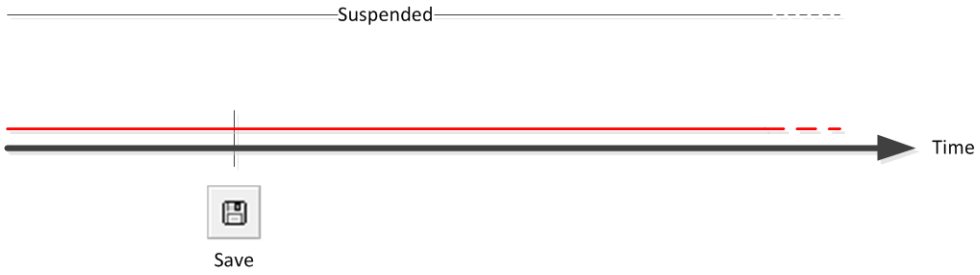


Figure 38: A suspended risk profile being held suspended.

When saving the risk profile it will remain suspended until a user later on decides to activate it manually (see Figure 38).

5.1.2.1.4 Date and Time for Later activation / suspension

Having selected the option of either activating or suspending a risk profile later, the ... button (also called a **Browse** button) opens a **Date Editor** dialog in which you can define the exact time of activation or suspension. For details please refer to section 4.3 for instructions about date and time.

5.1.2.2 Working with Recurrence Patterns

The system provides the option to define periods for when risk profiles will be active. This is done in the **Recurrence** box. Thus it is possible to define a pattern or several patterns for when a risk profile should be active such as e.g. every Monday between 18:00 and 19:00.

NOTE:

The recurrence patterns acts as additional conditions to the "active" state of the risk profile. A risk profile will only participate in the analysis process when the risk profile is active (as specified by the activate box above) and at least one of the specified recurrence pattern is matched by the current time. Also, the recurrence patterns have no effect when a risk profile is suspended!

If no recurrence patterns are specified the risk profile is entirely controlled by its "active" state.

Finally, risk profiles will appear to be active (on e.g. the profiles list) regardless of any matched or unmatched recurrence patterns.

The **Add** button (see Figure 24) is used for adding a new Recurrence Pattern, the **Edit** button (see Figure 24) for editing an existing Recurrence Pattern and the **Delete** button (see Figure 24) for deleting an existing Recurrence Pattern. The Recurrence Pattern that has been defined for a risk profile is listed in the text box below the **Pattern** radio button.

In order to define a new pattern click the **Add** button which opens the **Period selector** dialog, see Figure 39.

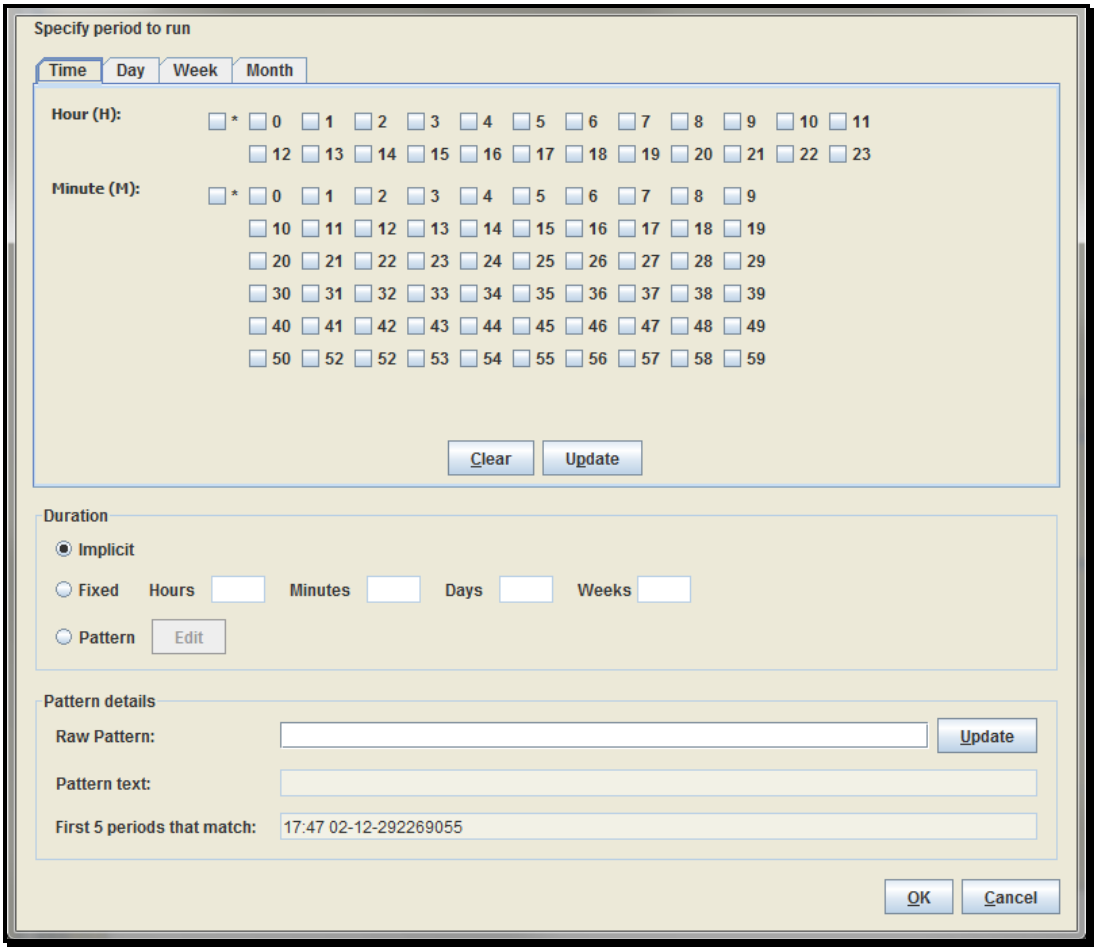


Figure 39: The Period Selector dialog

The upper part of the dialog contains four **Period** tabs for defining the schedule of when the risk profile should be activated, e.g. 10:00 every day or every Friday. The **Period** tabs are described in more details in section 5.1.2.2.1.

The purpose of the **Duration** box in the middle part of the dialog is to specify the duration of the risk profiles active period (**Implicit** or **Fixed**) or a set of periods for when the risk profile should be active.

The **Pattern details** area at the bottom of the dialog displays the details about the pattern that has been specified.

In order to set up a pattern the tick boxes are used in the tabs in order to set one or several activation times. Hereafter the relevant radio button is in the **Duration** area and the period or several periods are defined.

When finished working in the **Period selector** dialog click the **OK** button whereby the **Period Selector** dialog closes and the pattern text is displayed in the **Recurrence** box. If at a later point in time it is required to change or delete this pattern then select the pattern and then click the **Edit** or **Delete** buttons.

5.1.2.2.1 The Period Tabs

In the **Period** tabs you can specify a schedule for when a risk profile should be activated. There are four tabs to support the setting of time – the **Time** tab, the **Day** tab, the **Week** tab and the **Month** tab.

You can use any combination of the tabs to set one or multiple activation times. If you use them in combination it means that they will be added together meaning that the settings in each of the tabs must be fulfilled.

5.1.2.2.1.1 The Time tab

In the **Time** tab you define the hour and minutes for when a risk profile should be activated. In order to do so you tick the relevant tick box/boxes for setting hour/hours and minute/minutes. You can use either just the Hour tick boxes or the Minute tick boxes or you can use them in combination. If you use them in combination it means that they will be added together so if you e.g. tick the **Hour** tick box for the hour 12 and the **Minute** tick box for the minute 30 this means that the risk profile will be activated at 12:30 every day. Please observe that this is just the time when the risk profile is activated, the period for how long it should stay active is defined in the **Duration** box.

Besides tick boxes for each hour and each minute there is one tick box for hour and one for minute with a star to the right of them, . These boxes can be used to indicate every hour and/or every minute meaning that if you tick the one for hour the risk profile will be activated every whole hour, e.g. 16:00, 17:00 and so on.

You can use any combination of hours and minutes to set one or multiple activation times. When you are finished you click the **Update** button and the result will be displayed in the **Pattern details** box, refer to section 5.1.2.1.1.6 for further information. In Figure 40 below the risk profile will be activated every fifteenth minute every sixth hour, starting from the next coming sixth hour, e.g.:

0:00, 0:15, 0:30, 0:45,

6:00, 6:15, 6:30, 6:45,

12:00, 12:15, 12:30, 12:45,

18:00, 18:15, 18:30, 18:45.

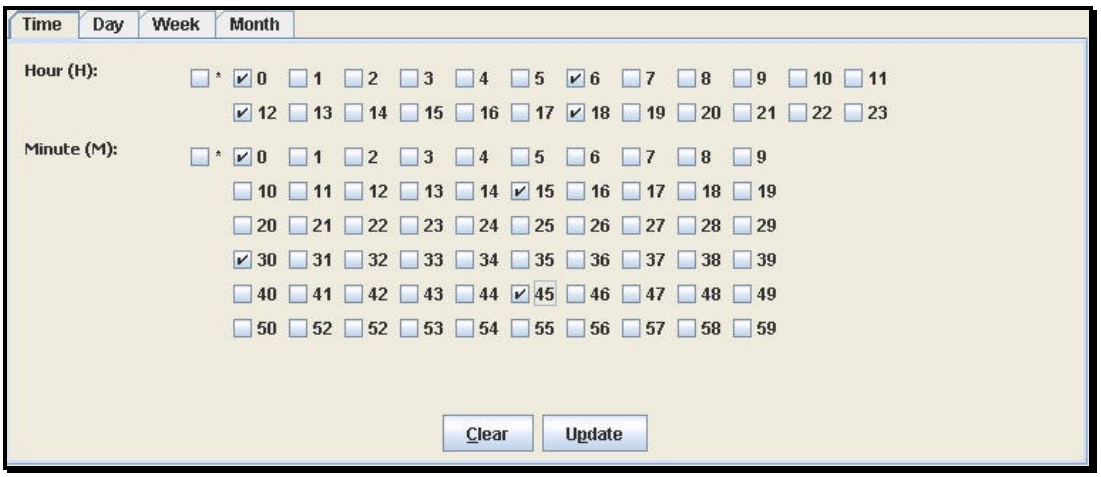


Figure 40: The Time tab

5.1.2.2.1.2 The Day tab

In the **Day** tab you define day or days for the activation of the risk profile. There are four options for use - three sets of tick boxes and one text box - and you can combine them as you wish. Please observe that this is just the time when the risk profile is activated, the period for how long it should stay active is defined in the **Duration** box.

The first option is the **Day of Week (DOW)** tick boxes and by ticking them you can set the activation time to any combination of day of week, e.g. Tuesday and Friday.

The next option is the **Day of Month (DOM)** tick boxes and by ticking them you can set the activation time to any combination of day of month, e.g. day 5, 10, 20 and 25.

The third option is the **Last Day of Month (DOM)** tick boxes and by ticking them you can set the activation time to any combination of the 10 last days of month, e.g. minus 2 days, minus 4 days and minus 6 days.

The last option is the **Day of Year (DOY)** text box and this can be used for setting one or several days of year. In order to do this you place the cursor in the box and then enter the day number from the keyboard. If you enter more than one day you must separate them with a comma and you must not enter any spaces in the string of days. Thus a correct string of days would be 25,50,100,343. The day numbers that are allowed are from 1 to 366 and they must not be negative, minus signs are not allowed.

Please observe that if you use the options in combination it means that they will be added together. If you e.g. tick the **Day of Week (DOW)** box for Friday and the **Day of Month (DOM)** box for 13 this means that the risk profile will be activated every Friday 13th – but not other Fridays than the 13th and not other weekdays on the 13th except Friday.

Besides tick boxes for each day of week and each day of month there is one tick box with a star to the right of them, . These boxes can be used to indicate every day of week and/or every day of month meaning that if you tick the one for day of week the risk profile will be activated every new day.

You can use any combination of days – both using the tick boxes as well as the **Day of Year** - to set one or multiple activation times. In Figure 41 below such a combination is used.

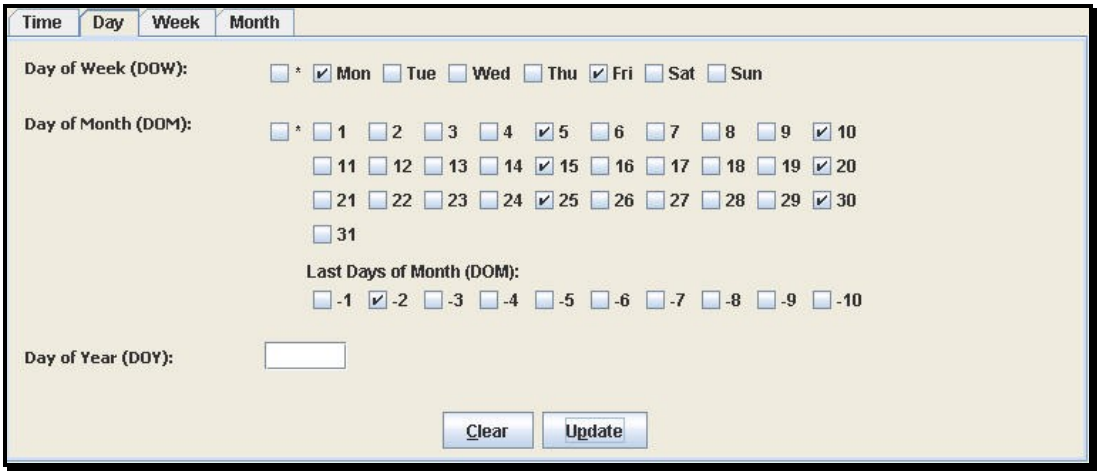

Time
Day
Week
Month
Day of Week (DOW):
* Mon Tue Wed Thu Fri Sat Sun
Day of Month (DOM):
* 1 2 3 4 5 6 7 8 9 10
11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30
31
Last Days of Month (DOM):
-1 -2 -3 -4 -5 -6 -7 -8 -9 -10
Day of Year (DOY):
Clear
Update

Figure 41: The Day tab with settings

5.1.2.2.1.3 The Week tab

In the **Week** tab you define week or weeks for the activation of the risk profile. There are two options for use - a set of tick boxes and one text box - and you can combine them as you wish. Please observe that this is just the time when the risk profile is activated, the period for how long it should stay active is defined in the **Duration** box.

The first option is the **Week of Month (WOM)** tick boxes and by ticking them you can set the activation time to any combination of week of month, e.g. week 2 and 4.

The second option is the **Week of Year (WOY)** text box and this can be used for setting one or several weeks of year. In order to do this you place the cursor in the box and then enter the week number from the keyboard. If you enter more than one week you must separate them with a comma and you must not enter any spaces in the string of days. Thus a correct string of weeks would be 5,25,30,43. The week numbers that are allowed are from 1 to 52 and they must not be negative, minus signs are not allowed.

If you use them in combination it means that they will be added together. If you e.g. tick the **Week of Month (WOM)** box for week 3 and in the **Week of Year (WOY)** box enter a string of weeks this means that the risk profile will be activated every combination of these weeks.

Besides tick boxes for each week of month there is one tick box with a star to the right of them, . This box can be used to indicate every week of month meaning that if you tick this box the risk profile will be activated every new week.

You can use any combination of weeks – both using the tick boxes as well as the **Week of Year** - to set one or multiple activation times. In Figure 42 below such a combination is used.

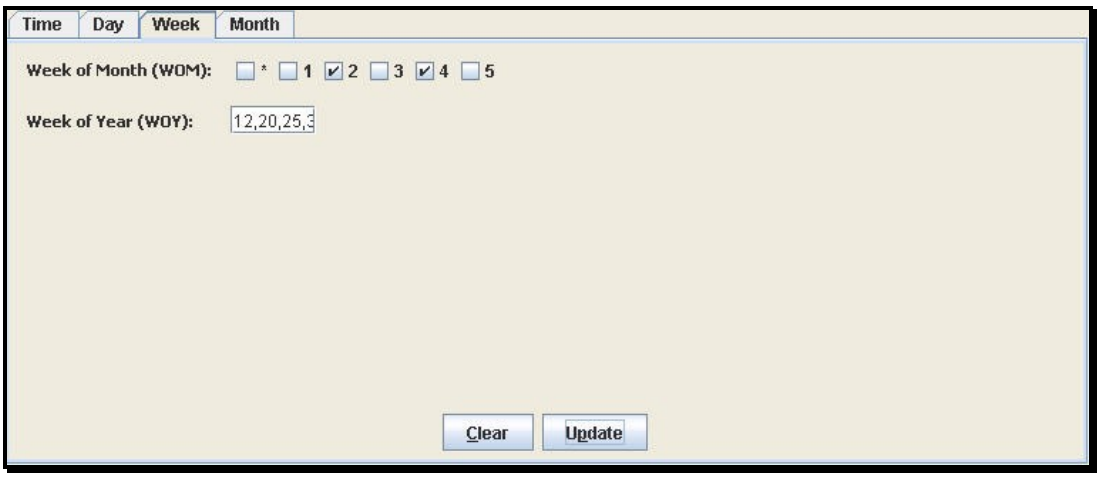


Figure 42: The Week tab with settings

5.1.2.2.1.4 The Month tab

In the **Month** tab you define month or months for the activation of the risk profile. There is a set of tick boxes for selection of month/months and you can combine them as you wish, e.g. every second month. Please observe that this is just the time when the risk profile is activated, the period for how long it should stay active is defined in the **Duration** box.

Besides tick boxes for each month there is one tick box with a star to the right of them, . This box can be used to indicate every month meaning that if you tick this box the risk profile will be activated every new month.

You can use any combination of months to set one or multiple activation times. In Figure 43 below such a combination is used.

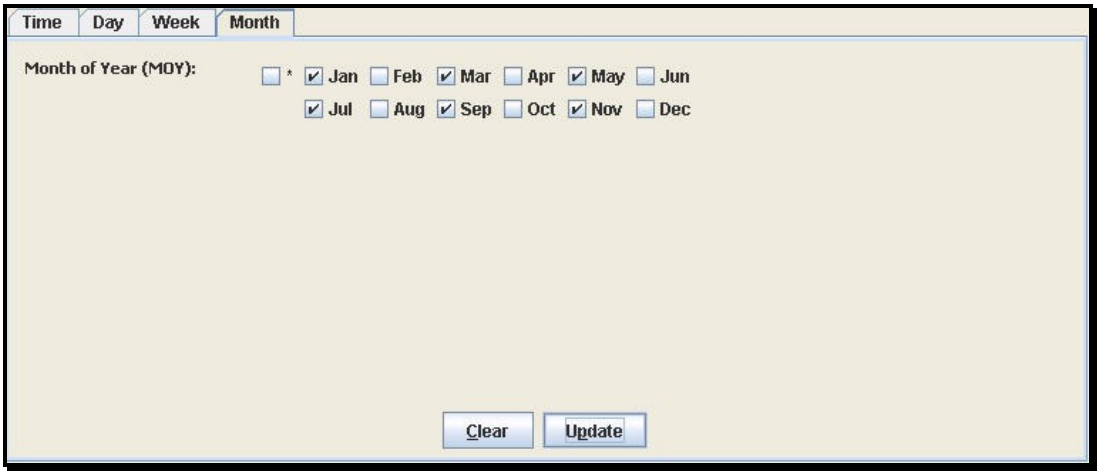


Figure 43: The Month tab with settings

5.1.2.2.1.5 The Duration Radio Box

The **Duration** box contains three radio buttons and these are used for specifying period or periods for when the risk profile should be active. By default the **Implicit** radio button is selected which means that the system deduct the period/periods based on the pattern settings in the Period tabs. If you e.g. set the activation time for a risk profile to 18:00 (by ticking the hour for 18:00 in the **Time** tab) the risk profile will be active for one hour between 18:00 and 19:00 every day. Please observe that the minimum duration for the current settings is indicated to the right of the **Implicit** radio button, in this case one hour. The length of the implicit duration will always match the smallest "period component" involved in the recurrence pattern. That is, if a pattern involves any of the minutes tick boxes from the Time Tab then the implicit duration will be one minute. If the least "period component" is from the Week Tab then a week will be the duration.

The **Fixed** radio button can be used to define a period in hours, minutes, days and weeks for which period the risk profile will be active starting from the activation time/times that you have defined in the Period tabs. If you e.g. in the Period tabs have set the risk profile to be activated every Friday at 18:00 and you have set a fixed duration of two hours the result will be that the risk profile will be active every Friday between 18.00 and 20:00, see Figure 44 below.

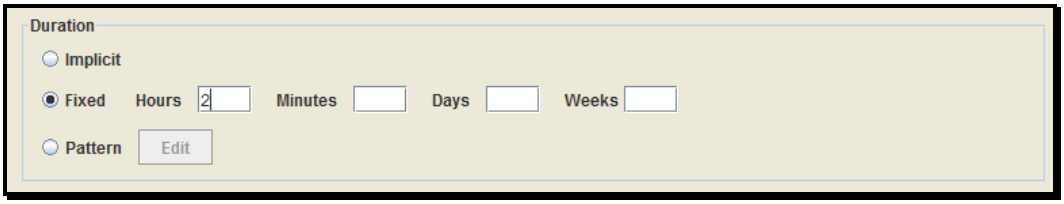


Figure 44: Illustrates a fixed duration of two hours

If you wish to define multiple periods for the risk profile to be active you can use the **Pattern** radio button. When you select the **Pattern** radio button the **Edit** button is enabled and when you click this button a dialog with Period tabs is opened. You can then use the tabs for defining the relevant period/periods. Refer to section 5.1.2.1.1 for information on how to set patterns.

As the **Time-**, **Day-**, **Week-** and **Month**-tab defines specific events (date and time) for when the risk profile shall become active defining a pattern in the **Duration** box will define specific events for when the risk profile shall become suspended.

Figure 45 shows an example of a complex scheduling for the first three month in the year 2010 having a pattern for activating the risk profile and another pattern for the duration.

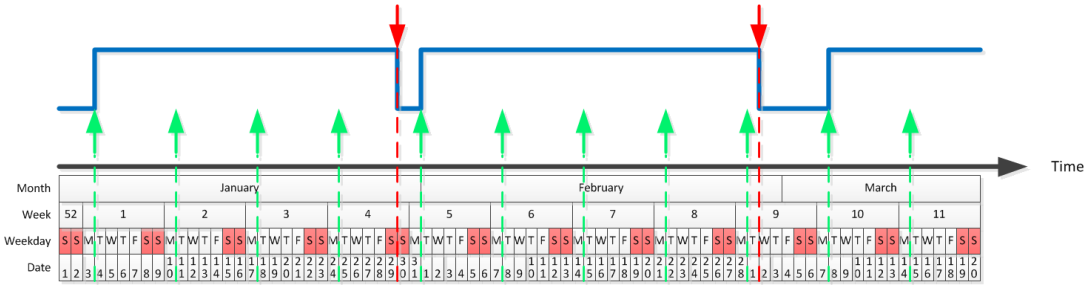


Figure 45: Example of a complex schedule.

The activation pattern activates the risk profile each Tuesday and the duration pattern suspends the risk profile at the second last day in each month. As shown in Figure 45 the activation pattern can be shown as a set of events (green arrows) as well as the duration pattern can be shown as another set of events (red arrows).

In the example the risk profile is suspended until the midnight between Monday the 3rd January and Tuesday the 4th January hereafter it will be active. The following midnights between the 10th and 11th, the 17th and 18th and the 24th and 25th of January can be ignore in fact the risk profile is already active.

Because Sunday the 30th of January is the second last day of the month the risk profile will be suspended at midnight between the 29th and 30th of January; however it will only remain suspended until Tuesday the 1st of February where activation pattern reactivates the risk profile.

A similar scenario takes place at the end of February; however the risk profile is suspended in more days.

5.1.2.2.1.6 The Pattern Details

The **Pattern details** box displays the current settings of the period(s) that you have defined in the Period tabs and in the Duration for when the risk profile will be active. These details are displayed in three boxes – the **Raw Pattern** text box, the **Pattern text** box and the **First 5 periods that match** box. Of these the **Raw Pattern** text box is editable while the others are read-only boxes. Every time you make an update in the Period tabs or the Duration box the information in the **Pattern details** box is updated.

The **Raw Pattern** text box displays minutes, hours, days and weeks that have been defined as activation time and period for the risk profile. The values in the text box can be changed by first clicking in the box and then keying in the relevant values. When you write the values you must follow a specific syntax for indicating minutes, hours, days, weeks and months and these are specified in the table below.

Recurrence Pattern:

Expression (*xx* indicate digits)	Means	Example	Explanation
M***m(,m)***	Minute	M10	Ten minutes past every hour
		M2,18	2 and 18 minutes past every hour
H***h(,h)***	Hour	H15	Three a clock PM
DOW***d(,d)***	Day of week	DOW1,3	Monday and Wednesday
		DOW6	Sunday
DOM***a(,a)***	Day of month	DOM25	The 25:th of every month
		DOM-3	The 3rd last day of every month
DOY***y(,y)***	Day of year	DOY315	Day number 315
WOM***w(,w)***	Week of month	WOM3	Week three of every month
WOY***v(,v)***	Week of year	WOY35	Week number 35
MOY***x(,x)***	Month of year	MOY6	June

where

m is an integer value between 0 and 59 expressing the minute within an hour

h is an integer value between 0 and 23 expressing the hour of a day

d is an integer value between 1 and 7 expressing the week day in the following way:

1 = Monday

2 = Tuesday

3 = Wednesday

4 = Thursday

5 = Friday

6 = Saturday

7 = Sunday

a is an integer value between -10 and 31 expressing the day on month relative to the start of the month.

-10 is the 10th last day of a month

-9 is the 9th last day of a month

…

-1 is the last day of a month

1 is the first day of a month

2 is the second day of a month

y is an integer value between 1 and 366 expressing the day of the year

1 is 1st of January

2 is 2nd of January

…

365 is the last day (31st of December in a non leap year)

366 is the last day (31st of December in a leap year)

w is an integer value between 1 and 5 expressing the week of a month

1 is the 1st week of a month

2 is the 2nd week of a month

3 is the 3rd week of a month

4 is the 4th week of a month

5 is the 5th week of a month

v is and integer value between 1 and 52 expressing the week of a year

1 is the 1st week of the year

2 is the 2nd week of the year

…

52 is the 52nd week of the year

x is an integer value between 1 and 12 expressing the month of a year

1 is January

2 is February

3 is March

4 is April

5 is May

6 is June

7 is July

8 is August

9 is September

10 is October

11 is November

12 is December

Duration pattern:

Expression (*xx* indicate digits)	Means	Example	Explanation
nW	Weeks	2W	Adds two weeks to duration
nD	Days	5D	Adds five days to duration

Expression (*xx* indicate digits)	Means	Example	Explanation
nH	Hours	4H	Adds four hours to duration
nM	Minute	13M	Adds thirteen minutes to duration

where ***n*** is an positive integer or 0.

In the text box the same expressions are used for defining the activation time(s) for a risk profile and the period(s) that the risk profile should be active. The first part is the activation time(s) and the second part is the period(s) and they are separated with plus sign (+).

If you wish to activate a risk profile at noon every Friday and run for 6 hours you must write the following in the **Raw Pattern** text box:

H12 DOW5 + 6H

When you push the update button this will be expanded into:

H12 DOW5 + 0W 0D 6H 0M

If you have used a tick box with a star ( ) the expression will be displayed in the **Raw Pattern** text box with a trailing star, e.g. DOM* which means all days in months. You can exchange the star with a relevant number if so needed. In the following example a risk profile will be activated at noon every day and stay active for three hours:

H12 DOW* + 0W 0D 3H 0M

When you are finished editing the values you click the **Update** button in the **Pattern details** box.

The **Pattern text** displays in plain words the settings for the risk profile in order for helping you to check the result of the current settings.

In order to provide you additional help to evaluate the result the **First 5 periods that match** box will display the first five periods that the risk profile will be active.

5.1.3 The Declaration Types sub-tab

Figure 46 below illustrates the Declaration Types sub-tab for a new risk profile. You tick the tick box for the declaration segment(s) (one or more) of the declaration type(s) (one or more) that the risk profile shall apply to. It is mandatory to select at least one segment in the Declaration Types sub-tab.

The combination of declaration segments/declaration types has the effect that it governs the declaration fields that will be available for defining conditions in the Risk Profile sub-tab. Only fields that are to be found in all selected declaration types and segments will be available.

Refer to section 4.2 Declaration Segments for further instructions.

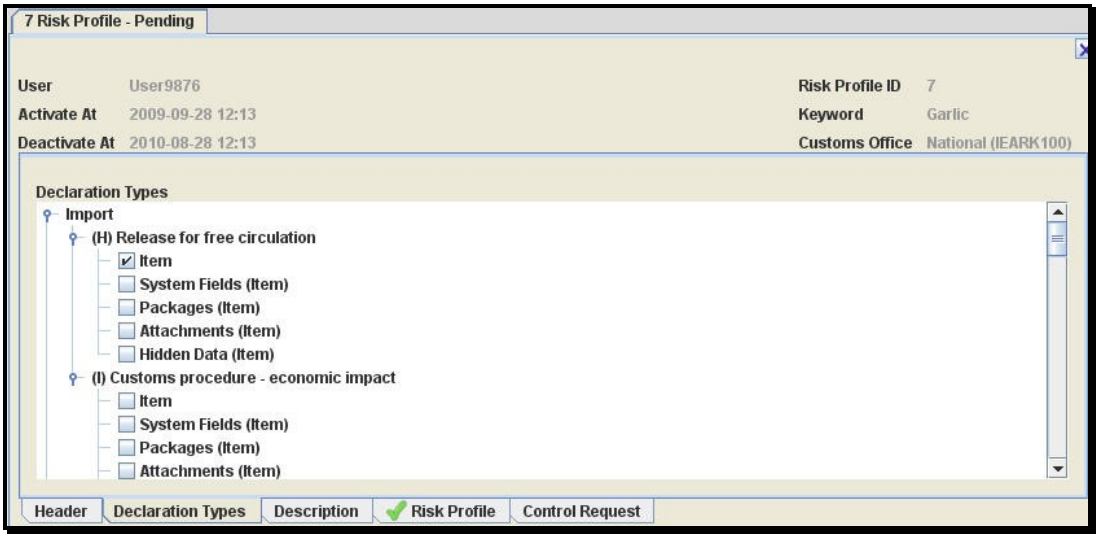


Figure 46: The Declaration Types sub-tab for a new risk profile

5.1.4 The Description sub-tab

The Description sub-tab contains two text boxes, a General description of Risk Profile box and a References box, see Figure 47 below. These serve the purpose of allowing you to record information about the risk profile such as the reason it is defined, what it is meant to trap, references to legislation or recommendations and other vital information. It is mandatory to enter a text in the General description of Risk Profile box but it is optional to enter text into the References box.

In either of the boxes it is possible to create hyperlinks to e.g. useful web-sites or to images on the Intranet, refer to section 4.8 for instructions about hyperlinks.

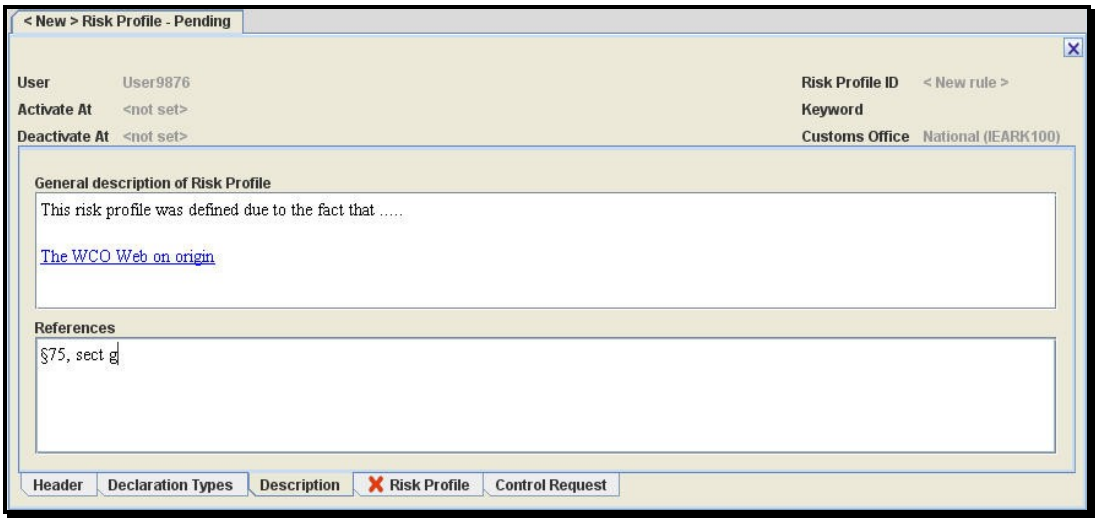


Figure 47: The Description sub-tab for a new risk profile

5.1.5 The Risk Profile sub-tab

The Risk Profile sub-tab is where you define the actual rules for risk profiles and this is done by preparing one or several conditions (also called partial expressions). Figure 48 below illustrates a new risk profile in this tab.

There are two methods for defining rules in the tab:

- using the Partial Expression View and/or
- using the Formalized Expression View.

The Partial Expression View is easy and intuitive to use but in this view it is not possible to define advanced rules.

The Formalized Expression View gives you full freedom of rule definition and supports the definition of advanced rules.

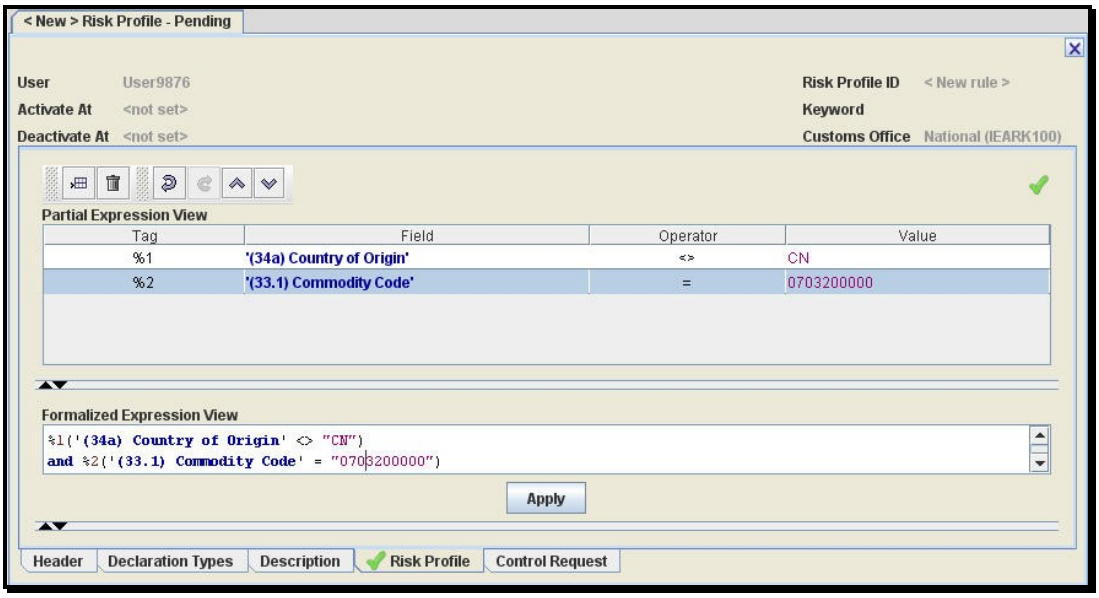


Figure 48: the Risk Profiles sub-tab for a new risk profile

A detailed description of how to work in the Risk Profiles sub-tab is to be found in section 5.7 Working with Risk Profiles Rules. For instructions on how to define rules refer to section 5.7.6 The Partial Expression View for the Partial Expressions View and section 5.7.7 The Formalized Expression View for the Formalized Expression View.

5.1.6 The Control Request sub-tab

The Control Request sub-tab is used for selecting the type of control to be carried out when a risk profile traps a declaration. You can define report back fields to be included in the inspection that will be carried out for the shipments that are trapped by the risk profile. Specific instructions to guide the officer who will perform the inspection can also be written. Figure 49 illustrates the Control Request tab for a new risk profile.

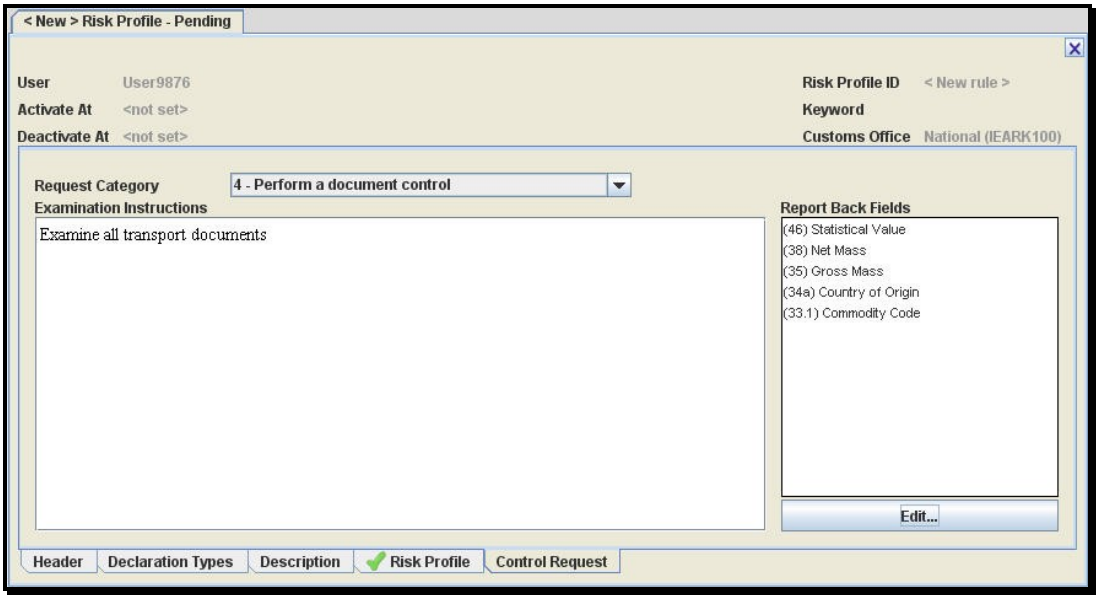


Figure 49: The Control Request sub-tab

5.1.6.1 Selecting Inspection Type

In order to select the inspection type to be carried out, you open the **Request Category** drop down box and select the appropriate inspection type. The following types are available:

0 - Perform a complete inspection
0 - Perform a complete inspection
1 - Perform a complete inspection and take a sample
2 - Perform a sample inspection
3 - Perform a sample inspection and take a sample
4 - Perform a document control

Figure 50: Selecting Inspection Type

5.1.6.2 Examination Instructions

The officers who shall perform inspections triggered by risk profiles might need some specific instructions on what actions they must take. When you define a risk profile it is possible to include these detailed instructions in the **Examination Instructions** text box. Figure 51 below contains an example of such instructions.

It is possible to create hyperlinks in the box and you will find further instructions on how to work with hyperlinks in section 4.8 Hyperlinks.

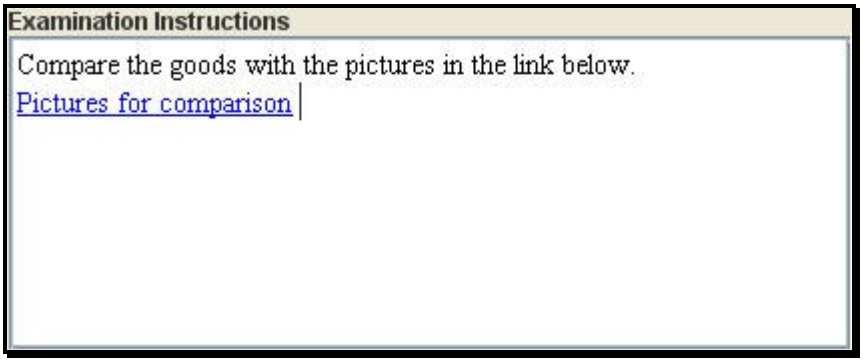


Figure 51: Examination Instructions

5.1.6.3 Selecting Report Back Fields

The purpose of selecting report back fields is to make it possible for the examining officer to report discrepancies between what is declared in the declaration and his findings from the inspection. When for instance an item in a SAD is trapped by a risk profile these fields will be included in the risk analysis result and will be sent to the declaration processing system for further preparation. To select report back fields, click the **Edit** button to open the **Select Report Back Fields** dialog. In this dialog you use the **Add, Remove, Add All** and **Remove All** buttons to select or de-select report back fields. Each field is listed together with its SAD-box number. It is also possible to use the standard Windows option of selecting a block of fields and selecting several non-consecutive fields.

If a block of fields is to be selected press the SHIFT key while clicking on the first and last field in the block. By pressing the CTRL key while clicking fields, non-consecutive fields can be highlighted.

Figure 52 below illustrates the process of selecting report back fields.

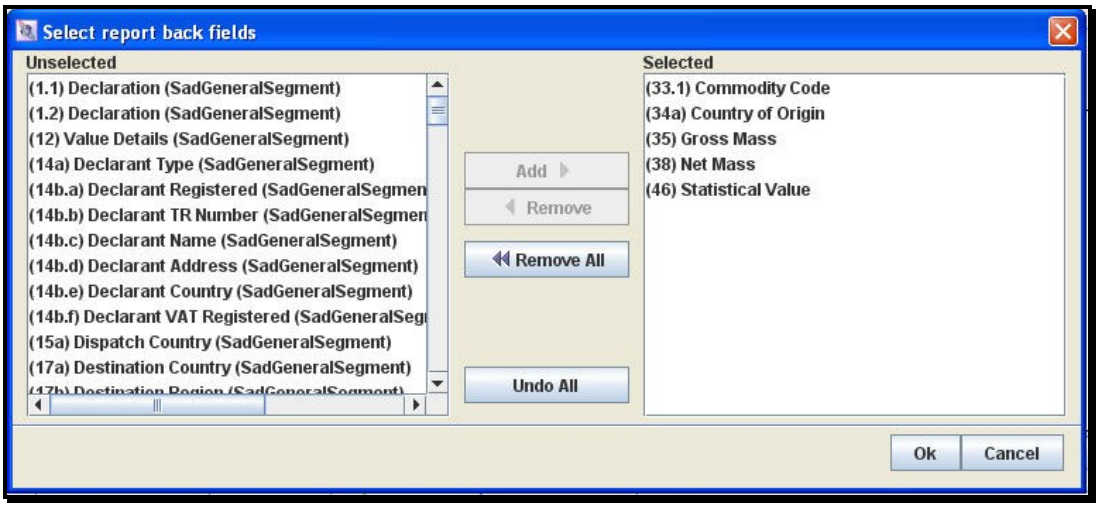


Figure 52: Selecting report back fields

☞ **Note:** The maximum number of fields that can be selected as report back fields is limited to six. If this limit is exceeded the system will display a warning.

☞ **Note:** Please note that the fields that are available depend on the declaration type(s) you have selected for the risk profile.

5.1.7 Saving the Risk Profile

The risk profile can be saved once it has been successfully defined. In order to do this you click the **Save** button, ,on the Toolbar. Alternatively you can select the **Save** menu item in the **File** menu or use the shortcut key combination CTRL-S.

5.2 Editing a Risk Profile

To edit an existing risk profile you must first locate it in the Table Area and if the list in the Table area is very extensive you may need to perform a filtering operation using the filter criteria in the Filter Area. If you know the keyword of the risk profile you can use that but if not you can use different combinations of filter criteria such as the user ID and a time interval. For further information on filtering operations refer to section 5.3 Filtering Risk Profiles.

When the risk profile to be edited is found in the Table Area you select it and click the **Open** button, , on the Toolbar. Alternatively you can select the **Open** menu item in the **File** menu or use the shortcut key combination CTRL-O. The risk profile is now open in read-only mode and to be able to edit it you click the **Edit** button, , on the Toolbar. Alternatively you can select the **Edit** menu item in the **Edit** menu or use the shortcut key combination CTRL-E.

You can now edit the details of the risk profile and for further information refer to section 5.7 Working with Risk Profiles Rules.

5.3 Filtering Risk Profiles

Risk profiles can be by using the filter criteria options in the Filter Area of the Risk Profiles tab. The area has three groups; the State group, the Declaration Types group and the Other group as can be seen in Figure 53 below. For further information on how to use the Declaration Types group the first five items in the Other group refer to section 4 General Configuration elements.

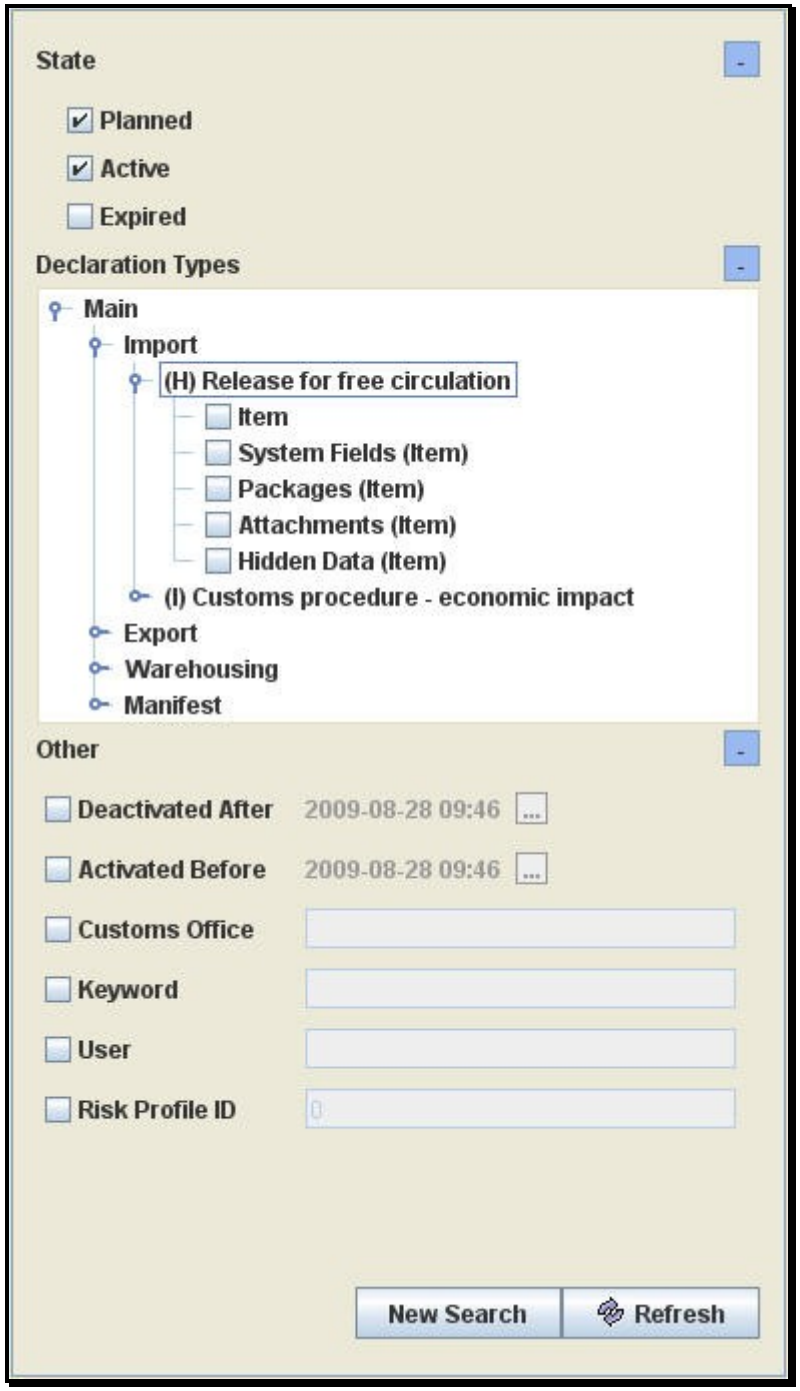


Figure 53: The Risk Profiles Filter Area

5.3.1 Filtering using the State Group

You can filter the list of risk profiles by using the different states for the risk profiles. In order to do so you tick the relevant tick box/boxes in the State area and then click the **Refresh** button.

5.3.2 Filtering using the Risk Profile ID Number

In order to use the Risk Profile ID Number for filtering you tick the **Risk Profile ID** tick box whereby the text box is opened and the cursor positioned to the left in the box. You then enter the risk profile ID number in the text box.

☞ **Tip:** Instead of ticking the tick box you can click in the text box and as a result the tick box is ticked and the text box is opened for edit.

5.4 Duplicating Risk Profiles

Instead of defining a new risk profile in it can sometimes be easier to duplicate an existing risk profile, make the necessary changes and then save it. For further information on how to duplicate refer to section 3.4.2.2 Duplicating Configuration Elements.

5.5 Risk Profile Revisions

A risk profile can exist in one or several revisions (or versions), refer to section 2.4 Revisions. All revisions are saved and stored in the system but by default only the latest revision of a risk profile is displayed in the Table Area and in the Data Area when it is opened. You can however open and work with older revisions of risk profiles and this can be useful e.g. when you wish to use the definitions in an older revision again. In order to work with older revisions for a risk profile you select the risk profile in the Table Area and the click the **Revisions** button, . Alternatively you can select **Show** Revisions from the **Edit** menu or use the short-cut key combination Ctrl-R. As a result the risk profile is opened in the Data Area together with a set of buttons for working with revisions, see Figure 54 below.

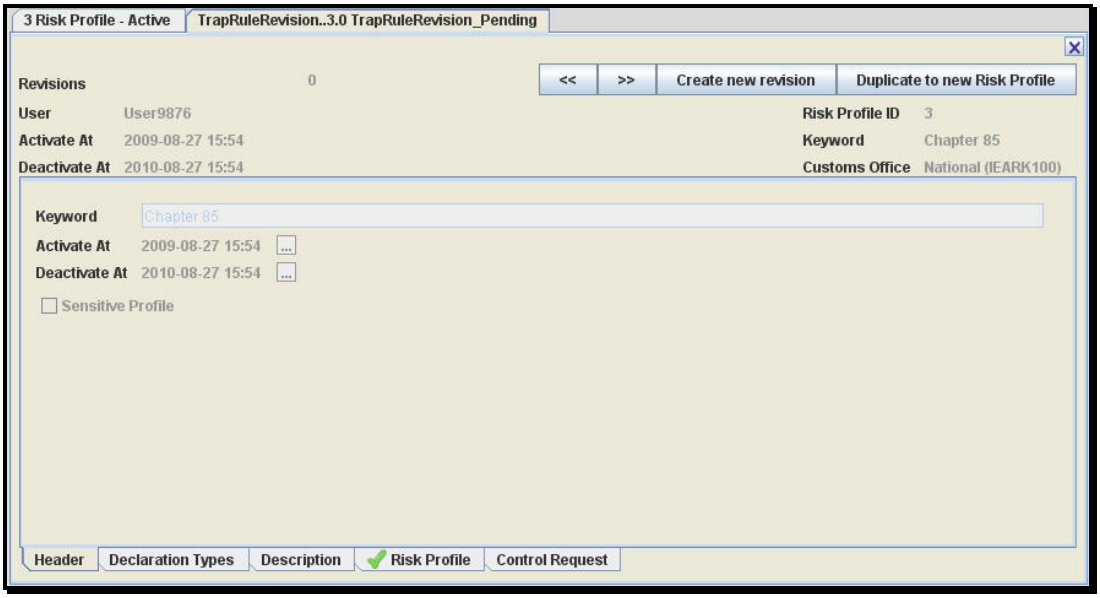


Figure 54: The revisions buttons

You can use the arrow buttons to scroll between the revisions for the risk profile in order to locate the appropriate one. In order to create a new revision you click the **Create new revision** button. A new revision of the risk profile is then opened in a separate tab in the Data Area in edit mode. You then make the necessary changes to the risk profile. For

detailed information about working with risk profiles refer to section 5 Working with Risk Profiles. When ready you click the **Save** button, , on the Toolbar. Alternatively you can select the **Save** menu item in the **File** menu or use the shortcut key combination CTRL-S. The risk profile is now saved with a new revision number.

Sometimes it might be useful to create a new risk profile based on an older revision for an existing risk profile meaning that you duplicate a certain revision. This can be done by clicking the **Duplicate to new Risk Profile** button. A new risk profile is then opened in a separate tab in the Data Area in edit mode. You then make the necessary changes to the risk profile. For detailed information about working with risk profiles refer to section 5 Working with Risk Profiles. When ready you click the **Save** button on the Toolbar. Alternatively you can select the **Save** menu item in the **File** menu or use the shortcut key combination CTRL-S.

5.6 Printing

Printing from the Risk Profile tab can be done in two different printouts – you can print the list of risk profiles in the Table Area or you can print risk profile details in the Data Area. For information about printing risk profiles refer to section 3.4.2.4 Printing Configuration Elements.

5.7 Working with Risk Profiles Rules

The work with risk profiles rules is performed in the Risk Profiles sub-tab. In this tab you have the possibility to define, edit and display risk profiles rules. The tab contains three views and two of these – the Partial Expression View and the Formalized Expression View – are used for creating risk profiles rules. The third view – the Validation View – informs you of the status of the two other views and will display warnings or error messages to assist you in your work. Figure 55 below illustrates the Risk Profiles sub-tab and its components.

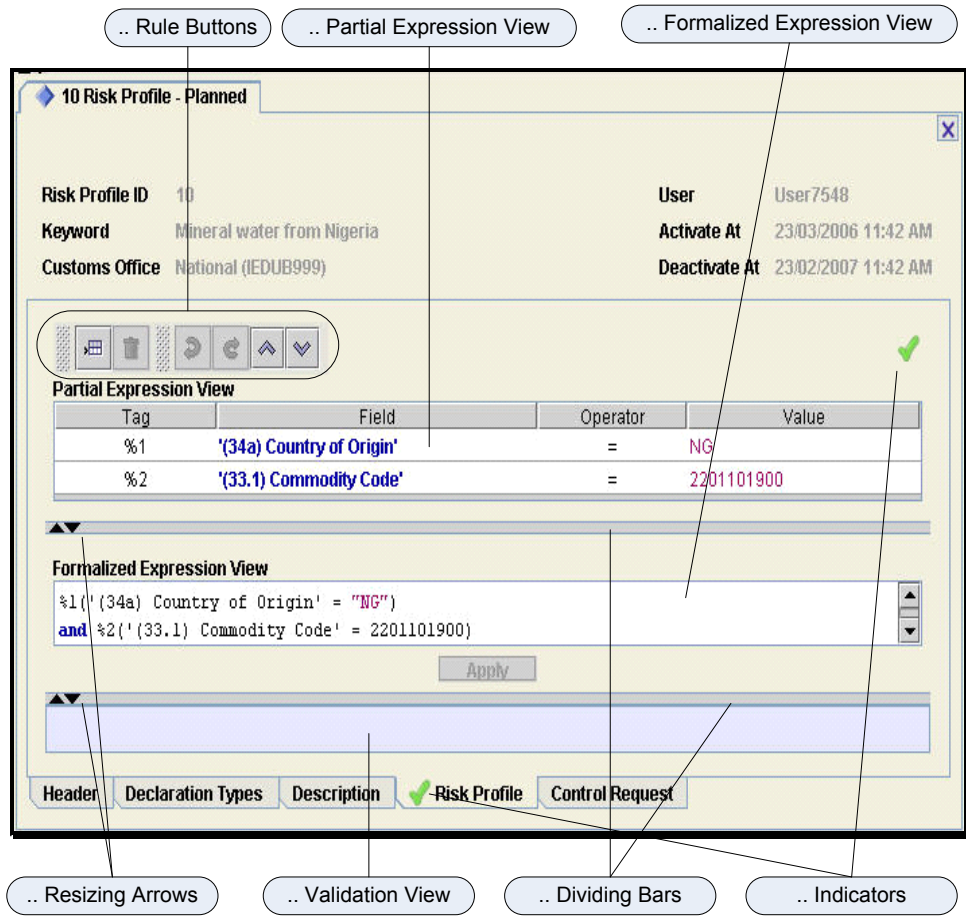


Figure 55: The Risk Profiles sub-tab

5.7.1 Re-sizing the Views

Each of the views can be re-sized whenever necessary, e.g. to make it possible to see all conditions in the Partial Expression View. There are three methods for use; dragging the **dividing bars,** clicking the **re-size arrows** or clicking the **Display** buttons.

5.7.2 Using the Dividing Bars to Re-size the Views

To re-size a view do as follows:

1. Move the cursor across one of the **dividing bars**
2. When the cursor changes shape to a double-pointed arrow, , drag the **dividing bar** in the desired direction

5.7.3 Using the Re-sizing arrows to Re-size the Views

To re-size a view do as follows:

1. To maximize a view click the **re-sizing arrow** that is pointing away from the view

2. To restore the size to normal click the **re-sizing arrow** that is pointing towards the view
3. To minimize a view click the **re-sizing arrow** that is pointing towards the view
4. To restore the size to normal click the **re-sizing arrow** that is pointing away from the view

5.7.4 Using the Display Buttons to Re-size the Views

To re-size a view do as follows:

1. To maximize the Partial Expression View click the **Show Partial Expression View** button
2. To maximize the Formalized Expression View click the **Show Formalized Expression View** button

5.7.5 Rule Buttons

The **Rule** buttons are used for defining and editing rules in the Risk Profile sub-tab views.

The buttons have the following actions:

Rule Button	Name	Action
	Insert Condition button	Inserts a new condition in the Partial Expression View
	Delete Condition button	Deletes a condition that you have selected in the Partial Expression View
	Undo button	Performs undo operations in the Partial Expression View and in the Formalized Expression View
	Redo button	Performs redo operations in the Partial Expression View and in the Formalized Expression View

Rule Button	Name	Action
	Display Formalized Expression View	Re-sizes the Formalized Expression View
	Display Partial Expression View	Re-sizes the Partial Expression View

5.7.6 The Partial Expression View

5.7.6.1 Columns

The Partial Expression View contains a table where each row represents a condition. The table contains the following columns:

- Tag
- Field
- Operator
- Value

The columns are described in the following table:

Column	Function	Description
Tag	Write protected	Contains a consecutive number of the conditions in the table, where the first (top) condition is tagged with % 1, the next % 2 etc. The field is controlled automatically by the system.
Field	Open at creation or in edit mode of planned risk profile, otherwise Write protected.	Drop-down box, that contains the declaration fields which are common for the selected declaration types.
Operator	Open at creation or in edit mode of planned risk profile, otherwise Write protected.	Drop-down box, that contains the operators, which are defined in the system, see section 5.7.9
Value	Open at creation or in edit mode of planned risk profile, otherwise Write protected.	Text field, where the user can enter the right side of the expression, e.g. a threshold, a value, etc.

The purpose for having Tags is to support linking of conditions between the Partial Expression View and the Formalized Expression View. Each tag is a unique identification of a condition which is administered by the system to ensure that the contents of the two views are synchronized.

5.7.6.2 Moving and Changing Columns

The columns in the view can be moved to suit your preference and facilitate your work. Columns are moved by dragging column header. As you move a column across another column the latter will jump to the opposite side of the column you are dragging. When you release the mouse button the column will then be placed in the position where the second column was placed before the moving operation. In the example in Figure 56 below the Field column is in the process of being dragged from a position to the right of the Tag column and the Tag column has just jumped to the right of the Field column.

☞ **Tip:** It is not necessary to drag a column all the way and adjust it in the vacant slot that the jumping column has created. As soon as the other column has jumped into position you can release the mouse button and the dragged column will automatically be adjusted into the vacant slot.

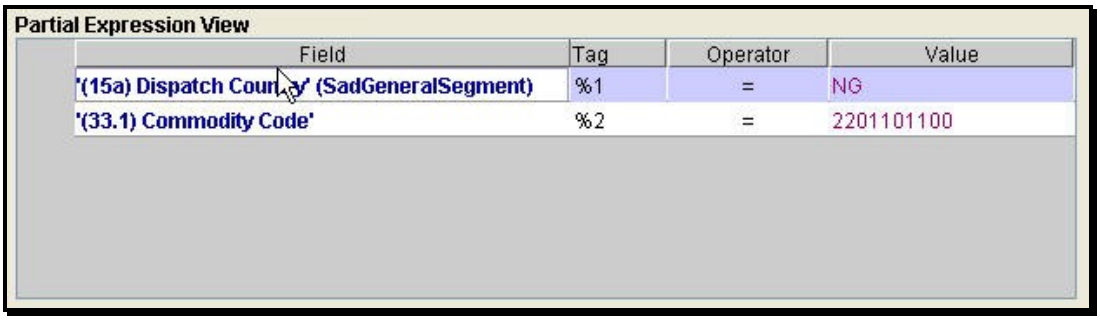


Figure 56: Moving columns in the Partial Expression View

The column width can be altered by placing the moving the cursor across the lines dividing the column headers. The cursor will then change shape to a double-pointed arrow . The column width can then be adjusted by dragging the cursor. In Figure 57 below the width of the Value column is in the process of being changed.

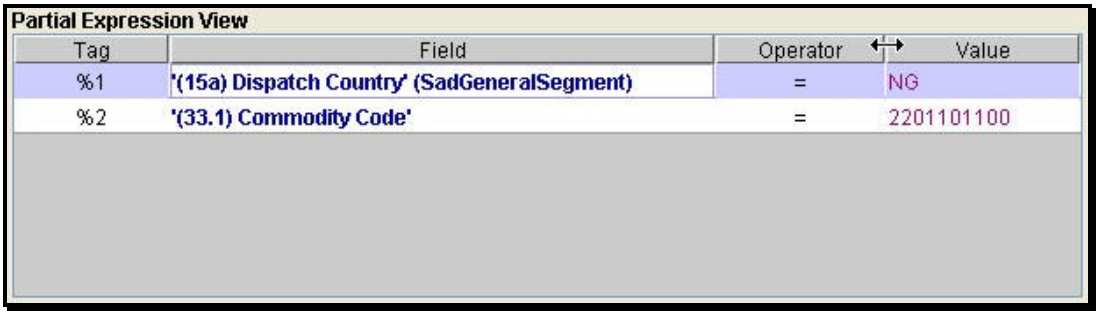


Figure 57: Re-sizing column in the Partial Expressions View

5.7.6.3 Setting Conditions

A Rule is created as a sequence of conditions which consists of a field, an operator together with a value. A condition expresses a logical value (true or false) and it can be set up in the following form:

Field Operator Value

The **Field** represents one of the declaration fields defined in the system, **Operator** defines the logical function between the left and the right side and **Value** represents a value entered by the user. A **Field** can e.g. be the *Net Mass,* **Operator** can be set as *less then* (<) and the **Value** can be *1000*. In this example condition would be

Net Mass < 1000

If the declared item net mass is less then 1000, the logical value of the condition is *true* – For all other values, it would be *false*.

When you start the process of defining a new risk profile the Partial Expression View is empty (as well as the Formalized Expression View). To set a condition you click the **Add partial expression** button, , whereby a new row is created in the Partial Expression View and the contents in the row is copied into the Formalized Expression View.

Figure 58: Inserting a condition

As can be seen in Figure 58 above the tag % 1 is assigned for the condition in both views in order to keep them linked together. The declaration field is the first from a list of the fields for the selected declaration type(s). You can change the declaration field by clicking on the field name whereby the **Field** drop down box opens and allow you to select another field. In Figure 59 below, the drop down box contains the fields that are available for selection and the user has selected the Commodity Code. As you click on a field, the drop down box will close.

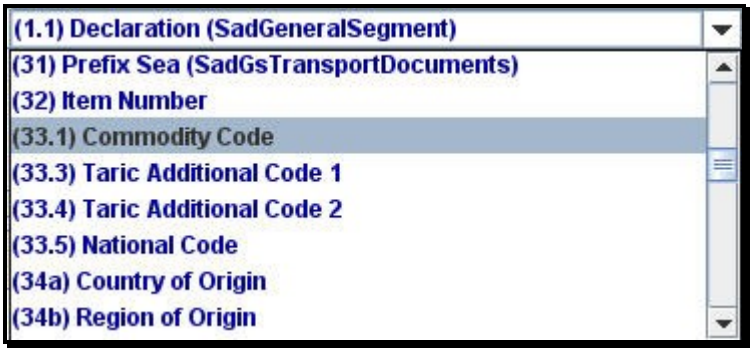


Figure 59: Selecting declaration field

The next step is to select an operator and to do this you click in the **Operator** drop down box, locate the relevant operator and click on it. As you click on an operator the drop down box will close.

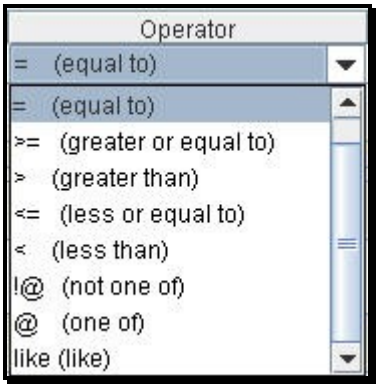


Figure 60: The Operator drop down box

For more details about the available operators refer to section 5.7.9 Operators.

The last thing to do in order to finish creating the condition is to enter a value. To do this you first click in the **Value** field, enter the relevant value and click anywhere in the Partial Expression View besides the value field or in the Formalized Expression View. As a result the value is copied into the value part of the condition in the Formalized Expression View.

You can now add more conditions and as you do so they are listed in numeric order. The number of conditions that can be added is unlimited.

5.7.6.4 Deleting, Undoing and Redoing Conditions

If you find that you need to delete a condition you first have to select it and then click the **Delete partial expression** button, . If you delete a condition other than the last the tags for remaining conditions will be renumbered.

Should you find that you made a mistake in deleting a condition you can easily bring it back using the **Undo** button, . The condition will be placed in its original position and the tags will be renumbered.

You also have access to a **Redo** button, , and using this together with the **Undo** button enables you to toggle deleted conditions.

☞ **Note:** As you delete, undo or redo conditions the same action takes place in the Formalized Expression View. If you e.g. delete a condition it will automatically be deleted in the Formalized Expression View.

5.7.7 The Formalized Expression View

The Formalized Expression View consists of a text box that can be utilized to write rules of more advanced nature. There are basically two ways to write the rules, either you work in combination with the Partial Expression View or you work in the Formalized Expression View only.

The conditions that you create in the view are continually validated to secure their compliance with the syntax language.

5.7.7.1 Writing Conditions

The conditions are written directly into the expression box and they must be entered in accordance with the syntax language that is implemented in the system. E.g. you wish to set a condition for a specific commodity code you enter the following condition:

'(33.1) Commodity Code'="5608111900"

The string ***(33.1) Commodity Code*** must be included in single quotation marks due to the fact that the long version of this fieldname contains special characters like spaces and braces. The short version of the field name is ***comm_code*** can also be used and in that case it does not need to be included in quotation marks. In this example the value inside the parenthesis is an indication to the box in the SAD where the commodity code is to be found. Since the commodity code field is of the alphanumerical data type (string), it is necessary to include the value inside double quotation marks. Values for fields of numerical data type (e.g. the statistical value) must not be inside quotation marks, see the example below:

'(38) Net Mass'>1000

In the Formalization Expression View it is possible to use both "**or**" and "**and**" for combining the conditions and you can also use calculations. For further information about the syntax language refer to2 Appendix 1.

It is also possible to use functions to facilitate writing conditions. E.g. if you are writing a condition where you wish to take a 20 random sample of goods with a specific commodity code you can use the $randomsample function, see the example below:

$randomsample(0.2) and '(33.1) Commodity Code'="5608111900"

For a complete list of available functions refer to Appendix 2

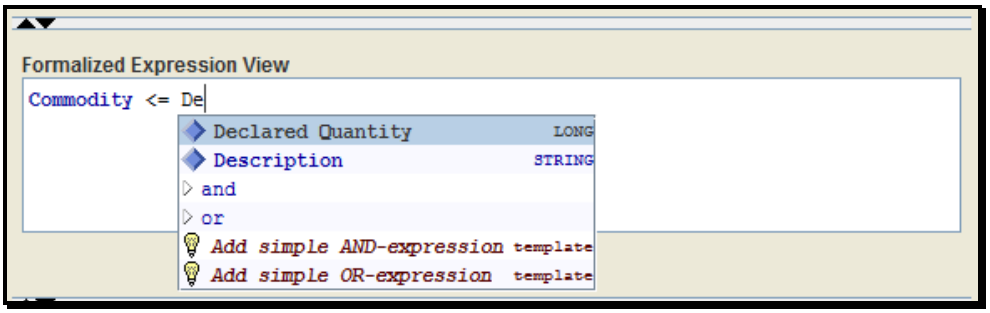
When your condition is completed the **Apply** button should be clicked and the synchronization with the Partial Expression View will be executed and the contents in the expression box will be checked for validity.

As previously mentioned you can either work with the Formalized Expression View in combination with the Partial Expression View or you can work in the Formalized Expression View only. If you opt to work in the Formalized Expression View only, you do not need to include any tags since there is no need for linking to the Partial Expression View. That is, the percentage-sign followed by a number (like `%2(my_field="some text")`) can be omitted in this case, and the expressions are no longer constrained to be on the form of a fieldname and an operator followed by a value.

5.7.7.2 The Auto Complete feature

When creating your conditions in the Formalized Expression View it is vital that fieldnames and function names are written without mistakes, in other words, that the spelling and upper/lower case must be correct. Luckily, the Formalized Expression View editor comes with a built-in "Auto Complete" feature that continuously offers suggestions on how to complete whatever you started out writing. Whenever activated, the "Auto Complete" list will bring up a list of options that either completes the word you are about to type or suggests the next token in your expression.

To activate the "Auto Complete" list you simply push "[**CTRL**] + [**SPACE**]" or point the mouse to the target and "**RIGHT-CLICK"**.



Also, when activated on an existing syntactically correct expression, the "Auto Complete" may be able to suggest alternatives for the token at the cursor position. That would for instance be the case when the cursor is placed on a fieldname in which case the complete list of valid alternative fieldnames is presented, that is, fields that can be accessed from the current context and are of the same type as the original field.

In order to expand the list with even more alternatives of say a different type, or from other parts of the configuration, simply repeat the activation of the "Auto Complete" list, that is, push [CTRL]+[SPACE] again. Depending on the context, the list may grow at each push for up to three activation pushes at which level all offered suggestions will be on the list;

On the initial level, after the first push, only the most obvious suggestions are on the list. If for instance you're in the process of writing a fieldname the list will usually be reduced to those fields that matches the fragment already typed and that belongs to the current context cube.

On the second push the list will grow to include all fields of all accessible cubes that match the fragment already typed.

On the third push the list will include fields of other types as well, if however, these where filtered out on the first hand, otherwise nothing happens on this push.

The "Auto Complete" list offers assistance to more than just fields and functions; it is also able to suggest cube names, dimension names, selected keywords and operators as well as entire templates like inserting a complete simple expression similar to what happens when adding a line in the partial expression view. The list displays a type name to the right of each suggestion in order to distinguish between the different kinds of suggestions. In this list a fieldname is referred to as a MEMBER since this is what it is actually called in the MDC language.

5.7.7.3 Using Aggregated Data

The following section requires a deeper insight of the multi dimensional cube language (MDC) and organization of the data structure of the analyzed data. It will, however, not go into deep details or try to explain all aspects and implications but rather strive to sketch out some of the possibilities and usages of "Anonymous Derived Members". Ordinary users may safely skip this part.

The following section is intended to describe the limitation when not using aggregated data, the reason for using aggregated data and how aggregated data is used.

5.7.7.3.1 Introduction

Declaration Structure

A typical declaration is structured in a hierarchical form, i.e. the declaration consists of a segment, which content is common for the whole declaration. A typical declaration also consists of one or more item segments, which are independent of each other but share the information in the common segment (also known as the header or general segment). Figure 61 shows a simplified example of such a declaration. In the example the declaration contains three items ($Item_1$, $Item_2$ and $Item_3$).

Limitations without Aggregation

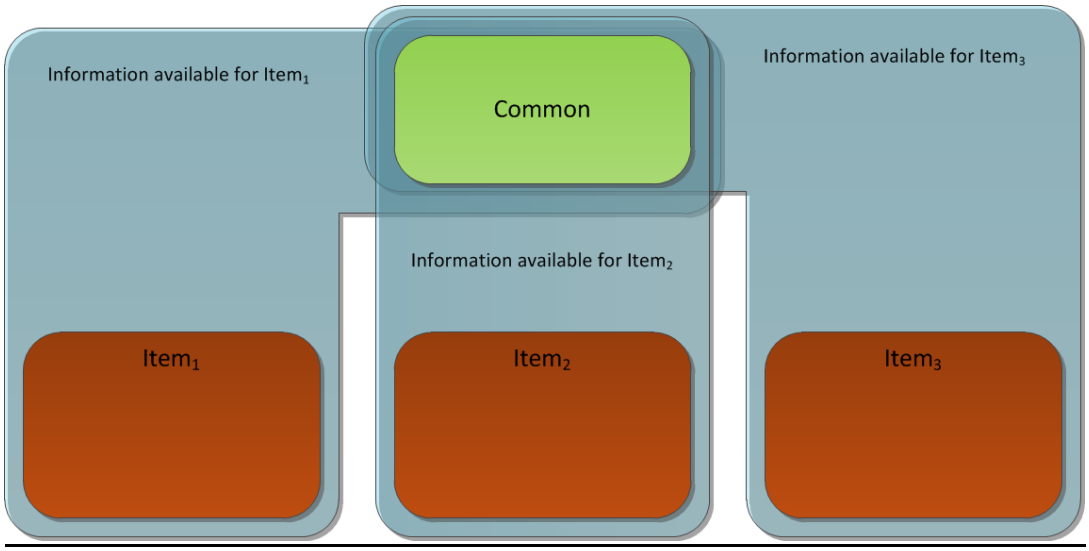


Figure 61: The item segments in a declaration are independent, but share a common segment of information.

Typically the risk analysis (as in the standard configuration) is configured to perform the risk analysis at item-level. This means that each item will be handled individually by the risk analysis process and will – during the process – be parsed through each relevant risk profile. In the example in Figure 61 the condition of any relevant risk profile will be evaluated three times – once for each item. When $Item_1$ is processed by a risk profile then only the information contained for $Item_1$ and the information contained in the common segment will be available for the condition of the risk profile; unless aggregated data is used it will not possible for the risk profile to take information from $Item_2$ or $Item_3$ into account.

We will now expand the example shown in Figure 61 with an additional level of information – In the example we call it the detail-level. This is shown at Figure 62.

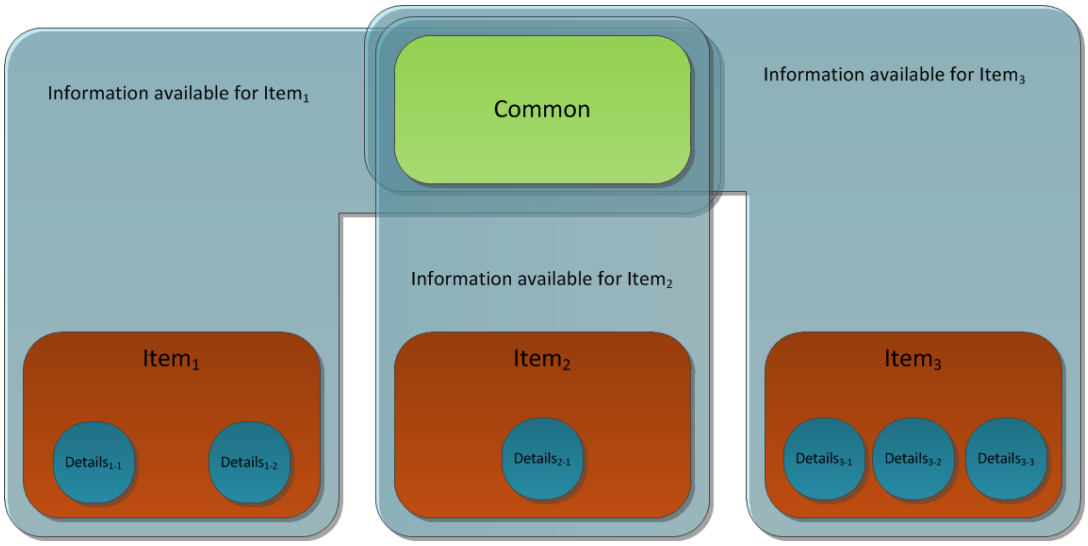


Figure 62: Example of a declaration with the detail-level included.

Figure 62 shows how each item-segment can contain an additional set of segments. In the example $Item_1$ contains two details-segments, i.e. $Details_{1-1}$ and $Details_{1-2}$, $Item_2$ contains one Detail-segment, i.e. $Detail_{2-1}$ and finally $Item_3$ contains three segments of Details: $Detail_{3-1}$, $Detail_{3-2}$ and $Details_{3-3}$. Because an item can contain an arbitrary number of Detail-segments then references from an item to a measure in the detail-level is ambiguous. E.g. if the Detail-type defines the field *A* then references from $Item_3$ to the field *A* will become an ambiguous reference due to the fact the it could be the field in either $Details_{3-1}$, $Details_{3-2}$ or $Details_{3-3}$.

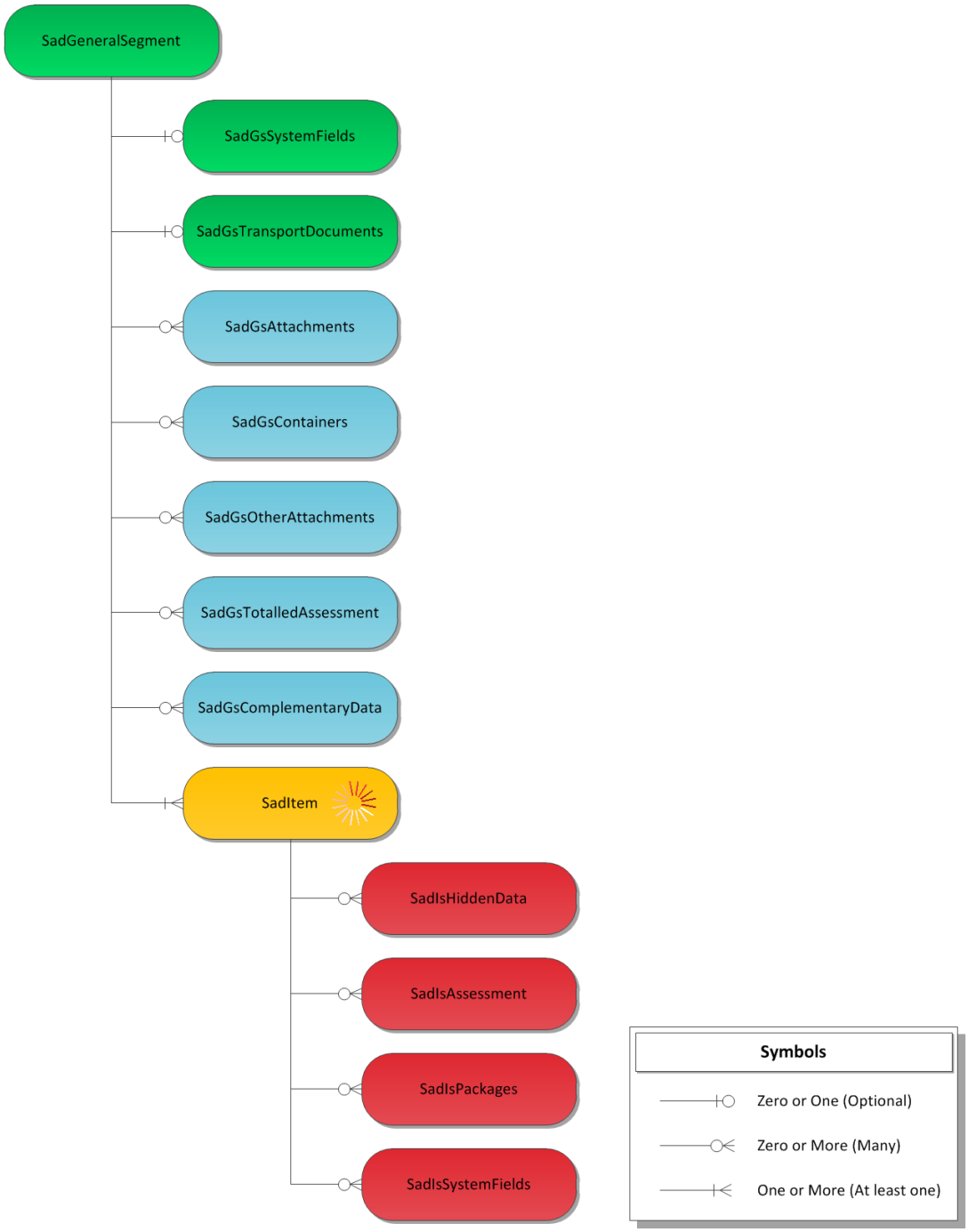


Figure 63: Hierarchical Structure in the SAD configured in the Standard System.

Figure 63 shows the hierarchical structure of the SAD in the standard configuration as an ER-diagram.

The diagram shows the *SadGeneralSegment* as the top level segment of data – also known as the header or general segment. Below the *SadGeneralSegment* exists the *SadGsSystemFields*- and *SadGsTransportDocuments*-segments, which are optional; this means that their contents can be included (or interpreted as if) in the common segment.

According to Figure 63 there can exists multiple *SadGsAttachments*-, *SadGsContainers*-, *SadGsOtherAttachments*-, *SadGsTotalledAssessment*- and *SadGsComplementaryData*-segments in each declaration, hence a reference to a field in one of these segments from the *SadItem*-segment will be ambiguous.

Below the *SadItem*-segment multiple *SadIsHiddenData*-, *SadIsAssessment*-, *SadIsPackages*- and *SadIsSystemFields*-segments can exist, hence a reference to a field in one of these segments from the *SadItem*-segment will also be ambiguous.

The conclusion should be that unless aggregated data is used then a risk profile running on the item level will only be able to address fields in the *SadGeneralSegment*, *SadGsSystemFields*- and *SadGsTransportDocuments*-segments as well as the current *SadItem*-segment it self.

Cubes and Coordinates

In order to understand how aggregated values can be used to make risk profiles address fields inside segments which are otherwise unreachable, we need to understand the concept of cubes and coordinates.

In ESKORT each segment-type (or entity-type) shown in Figure 63 is modelled as cubes. A cube is a structure of data multidimensional coordinate system.

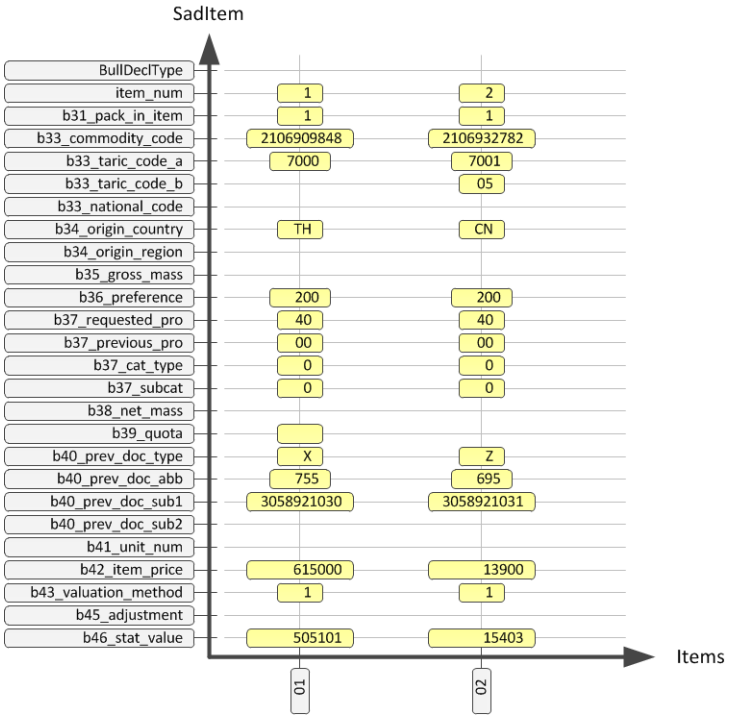


Figure 64: Example of the contents of the SadItem-cube.

Figure 64 shows an example of the contents of the *SadItem*-cube. The cube is spanned by two dimensions:

Items: A dynamic dimension having a value for each item contained in the declaration. The dimension is dynamic, which means that the number of values on the *Items*-axis depends on the number of items contained in the declaration, which can differ from declaration to declaration.

SadItem: A static, measure dimension of the cube. The dimension contains a fixed set of fields (as labels on the *SadItem*-axis).

Each combination of one value on from each of the axis' defined by the cube constitutes a coordinate. In the example shown in Figure 64 this will be a value from the *Items*-axis and a value from the *SadItem*-axis, e.g. '02' from the *Items*-axis and 'b34_origin_country' from the *SadItem*-axis. The simplest way the coordinate can be expressed is on the following form:

```
{ '02', b34_origin_country }
```

Figure 65: Simplest form expressing a coordinate.

If required the coordinates can extended to also include the name of dimension or/and the name of the cube. This is shown in the examples in Figure 66 and Figure 67.

```
{ '02', SadItem:b34_origin_country }
```

Figure 66: Coordinate where the measure is specified with the cube name.

```
{ Items.'02', SadItem:b34_origin_country }
```

Figure 67: Coordinate where the measure is specified with the cube name and the value on the Items-dimension is prefixed by the dimension name.

As shown in Figure 64 each coordinate points out a cell in the cube. The coordinate`{'02', b34_origin_country}` points out the cell containing the value *CN* – we call *CN* the value of the cell at the coordinate.

Some coordinates points out an empty cell, e.g. in the example shown in Figure 64 the coordinate`{'01', b38_net_mass}` points out an empty cell. We then call the value on that coordinate for *Unknown*, which can be interpreted as if no value was declared for that field.

5.7.7.3.2 Aggregating Data

Aggregation of data is a way of making risk profiles use information from another cube than the iteration cube and which consists of more dynamic dimensions than the iteration cube. In the example from Figure 63 this means letting a risk profile which iterates on the *Items*-dimension (i.e. parsing through each *SadItem*-segment in a declaration) e.g. access values contained in the *SadIsPackages*-segments.

Basically aggregation is made by performing some kind of operation of the ambiguous fields in a cube. As an example let us take a look at the *SadIsPackages*-cube.

The *SadIsPackages*-cube contains one more dynamic dimension than the *Items*-dimension defined for the *SadItem*-cube – i.e. the *IsPackages*-dimension.

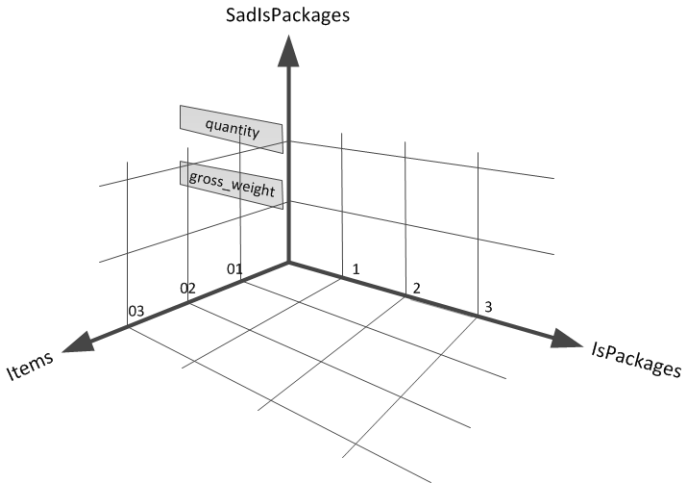


Figure 68: Illustration of the SadIsPackages-cube.

Figure 68 shows that the *SadIsPackages*-cube consists of the static, measure dimension: *SadIsPackages*. For simplifying the drawing in Figure 68 only shows a subset of the measures – i.e. *quantity* and *gross_weight*. The *SadIsPackages*-cube also consists of the dynamic *Items*- and *IsPackages*-dimensions. Due to the 3 dimensions any cell value must be addressed by a coordinate containing three values. On Figure 69 this the addressing of a cell is illustrated by the coordinate `{ '02', '3', gross_weight }`.

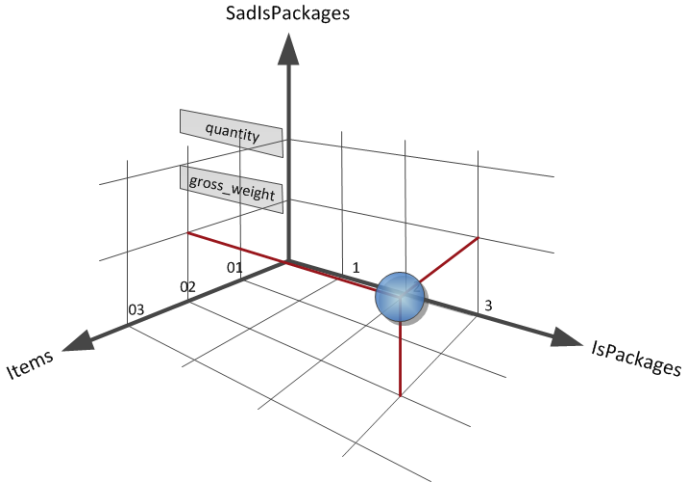


Figure 69: Cell pointed out by a coordinate.

From a risk profile's (which iterates on the *Items*-dimension) point of view each iteration (i.e. each item on the declaration) is handled independently, hence we can ignore the *Items*-dimension in fact each iteration on the *Items*-dimension can be illustrated as another projection is being made in the cube bringing the three-dimensional *SadIsPackages*-cube into two dimensions.

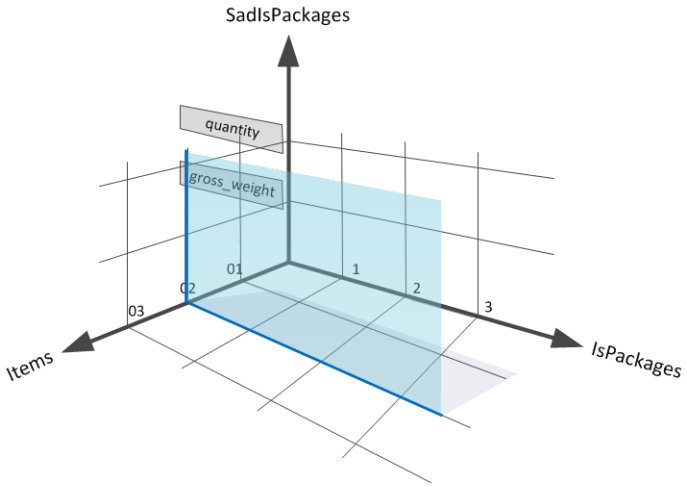


Figure 70: SadIsPackages-cube being projected into two dimensions.

Figure 70 illustrates the projection being made for the iteration where the *Items*-dimension is being given the value '02'. After the projection each iteration (e.g. '02') in the cube will look as in Figure 71.

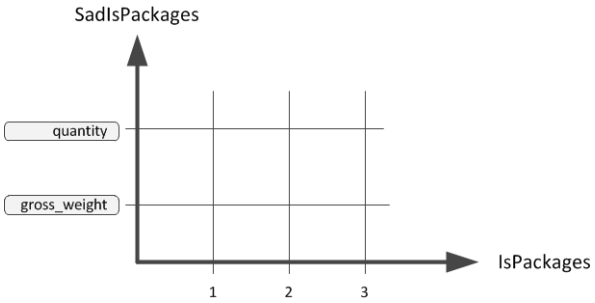


Figure 71: The projected SadIsPackages-cube.

The projection also reduces the number of values that must be contained in coordinate from three to two – this is illustrated at Figure 72.

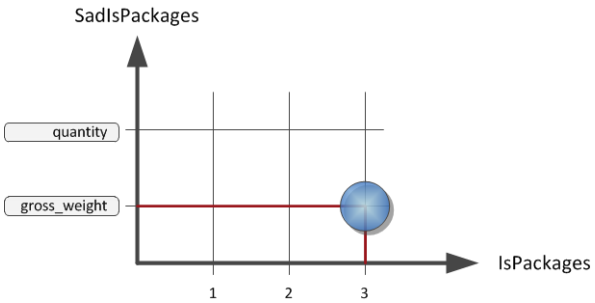


Figure 72: After projection the coordinate only needs two values.

In Figure 72 the coordinate has been reduced to `{ '3', gross_weight }`.

Even though we have reduced the *Items*-dimension from the *SadIsPackages*-cube from the *SadItem*-cube's point of view the *IsPackages*-dimension in the *SadIsPackages*-cube is still the reason for the ambiguous fields, i.e. the *quantity*- and the *gross_weight*-fields can have

different values for each value on the *IsPackages*-dimension. To be able to address these fields from the context of a *SadItem*-segment the field values must be aggregated.

In ESKORT the following aggregation functions are available:

SUM(*Dim*)	The *SUM*-function adds all numeric values in the specified dimension together and provides the total as the result of the function.
AVG(*Dim*)	The *AVG*-function calculates the average of all numeric values in the specified dimension.
MIN(*Dim*)	The *MIN*-function determines the minimum value of all numeric values in the specified dimension.
MAX(*Dim*)	The *MAX*-function determines the maximum value of all numeric values in the specified dimension.
SET(*Dim*)	The *SET*-function combines all the values contained in the specified dimension into a set.
AGGREGATE(*Dim*)	The *AGGREGATE*-function uses the aggregation-function defined for the specified dimension.

In a coordinate the functions can be used to substitute the value for a dimension (like the *IsPackages*-dimension), hence the no specific value in the *IsPackages*-dimension needs to be specified, but just the aggregation. Looking at Figure 72 we no longer need a specific value for the *IsPackages*-dimension but we can represent the dimension by an aggregation function.

```
{ IsPackages::SUM( IsPackages ), SadIsPackages:gross_weight }
```

Figure 73: Coordinate to SadIsPackages containing an aggregation.

When examining the coordinate in Figure 73 we see that it is made by an aggregation and a field.

The aggregation is the SUM-function which is adding up the values over the *IsPackages*-dimension. The field is the gross weight; this provides us the sum of all gross weights in the *SadIsPackages*-cube for the item in question. The first impression of the coordinate in Figure 73 looks quite complex; however it can be explained by an example.

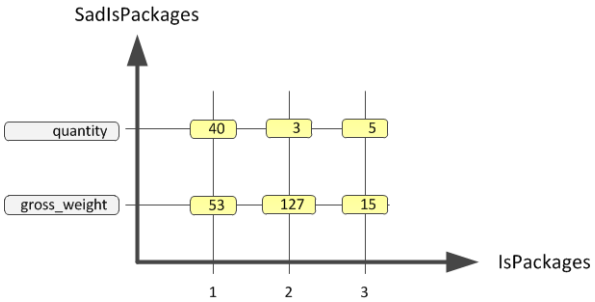


Figure 74: Example of contents of the SadIsPackages-cube for a given SadItem.

Figure 74 shows an example of the contents of the *SadIsPackages*-cube for a given item.

The first part of the coordinate (`IsPackages::SUM( IsPackages )`) adds a new (derived) member to the cube; the member is expressed by the SUM-equation, but remains anonymous. As shown in Figure 75 the new derived member can be interpreted as a new column `SUM( IsPackages )`.

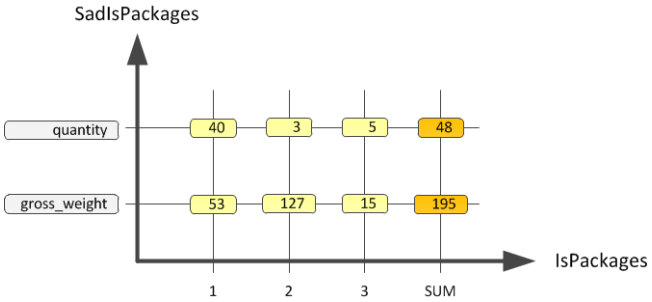


Figure 75: The aggregation introduces a new column.

The value of the cell on the coordinate

```
{ IsPackages::SUM(IsPackages), SadIsPackages:gross_weight }
```

will then be the sum of all the gross weight for the item in question, this is: 195.

To express the sum of the quantity the *quantity* member shall be referred instead of the *gross_weight*. The coordinate will look like this:

```
{ IsPackages::SUM(IsPackages), SadIsPackages:quantity }
```

For the *SadItem* in question the value in the cell at the coordinate will be 48.

Now it is easy to add additional coordinates using other aggregation functions such as MAX. Figure 76 shows how this can be made for the *quantity*-field.

```
{ IsPackages::SUM(IsPackages), SadIsPackages:quantity }
```

Figure 76: Coordinate using MAX as aggregation function.

This can be visualized as shown in Figure 77.

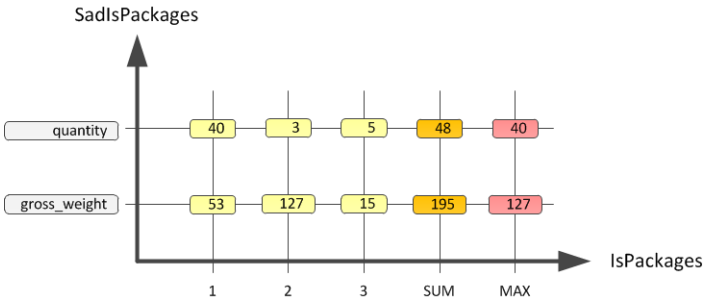


Figure 77: The SadIsPackages-cube with the SUM and MAX aggregation.

Please note that the aggregation function does not take the result of other aggregations into account, i.e. in Figure 77 the MAX-function does not take the calculated values from the SUM-function into account. E.g. for the *gross_weight* the MAX-function calculates the maximum as being the maximum values of the original (yellow) values, i.e. 53, 127 and 15.

5.7.7.3.3 Using Calculations in Aggregations

In the previous section we saw how aggregation could be made by using aggregation functions such as SUM and MAX. Many situations can be covered by using these functions; however in other situation we need to do some more calculations before we make the aggregation.

Let us take offset in the *SadIsPackages*-cube (shown in Figure 74) once more. Let us say that we want to count the number of packages for a given item which gross weight is more than 20 kg. Hereafter the result shall be used in the condition of a risk profile, so if there are more than 3 packages for the item in question having a gross weight more than 20 the risk profile condition shall evaluate to true.

This can be done by using the coordinate shown in Figure 78.

```
{
  IsPackages::SUM( IsPackages ),
  SadIsPackages::iif( SadIsPackages:gross_weight > 20, 1, 0 )
}
```

Figure 78: Coordinate which represents the number of packages which gross weight is more than 20.

The second value in the coordinate in Figure 78 now contains a more complex calculation.

The calculation is using the *iif*-function, which contains three arguments. The first argument is an equation which shall evaluate a boolean value (either true or false); if the first argument evaluates to true the *iif*-function returns the second argument; otherwise the third argument is returned. In the coordinate in Figure 78 the *iif*-function tests whether the gross weight is more than 20 kg; if that is the case 1 is returned otherwise 0.

You can think of the statement as if a new, calculated measure is introduced into the *SadIsPackages*-cube. This is illustrated in Figure 79. The new measure is being given a name, hence it is anonymous.

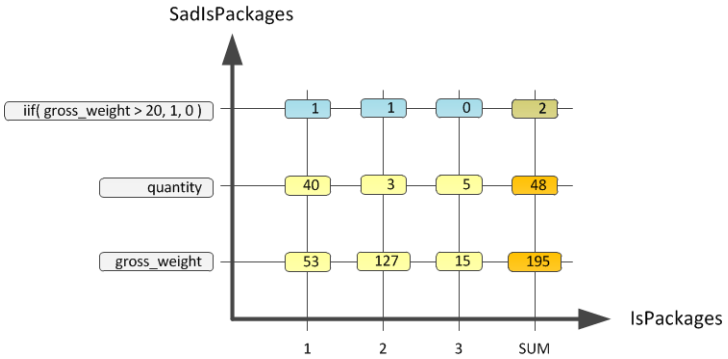


Figure 79: The SadIsPackages cube has been extended with a calculated measure.

Because the coordinate shown in Figure 78 also uses the `SUM(IsPackages)` the coordinate will point at the cell which represents the sum of all the measures calculated by the *iif*-function; this will be the number of *IsPackages* having a gross weight more than 20kg, which are 2.

Figure 80 shows how the condition in a risk profile can use the coordinate.

```
{
  IsPackages::SUM( IsPackages ),
  SadIsPackages::iif( SadIsPackages:gross_weight > 20, 1, 0 )
} > 3
```

Figure 80: Condition for the Risk Profile

In the example used so far the condition will evaluate to false due to the fact that only two IsPackages have a gross weight of more than 20kg and at because more than three are required to make the condition evaluate to true.

5.7.7.3.4 Example 2 – Maximum rate between gross mass and quantity

This section is shows another example of how anonymous derived members can be used to solve a problem.

Let us say we want to make the condition for a risk profile which fires when the maximum rate between the gross weight and quantity in the SadIsPackages for the item is more than 10. We do that by the condition shown in Figure 81.

```
{
  IsPackages::MAX( IsPackages ),
  SadIsPackages::(
    SadIsPackages:gross_weight / SadIsPackages:quantity
  )
} > 10.0
```

Figure 81: Example 2 – Condition for risk profile.

As shown in Figure 82 the condition makes a new calculated measure (gross weight / quantity) and finds the maximum value of the calculation from the *IsPackages*, which are 42.33. Because 42.33 are more than 10 the condition evaluates to true.

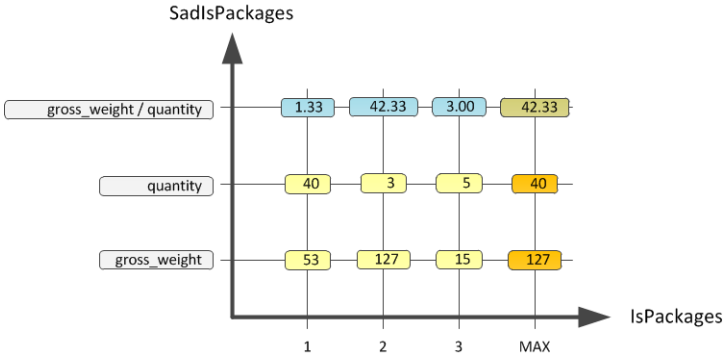


Figure 82: The maximum rate between the gross weight and the quantity is determined.

5.7.7.3.5 Example 3 – sad

We would like to make the condition for a risk profile which runs on each item. The risk profile shall fire if the following conditions are fulfilled:

1. The first four digits in the commodity code shall be "1234".
2. The total sum of all gross weights in the *SadIsPackages*-cube shall be more than 10kg.
3. No required attachments must have been declared having the certificate type "ABC".

The condition is shown in Figure 83.

```
  '(33.1) Commodity Code' like "1234*"
and
{
  SadIsPackages::SadIsPackages:'(31) Gross Weight',
  IsPackages::SUM( IsPackages )
} > 10.0
and
{
  SadGsAttachments::(
    iif(
        SadGsAttachments:'(44) Certificate Type' = "ABC"
      and
        SadGsAttachments:'(44) Required' = "yes",
      1,
      0
    )
  ),
  GsAttachmentItems::SUM( GsAttachmentItems )
} = 0
```

Figure 83: Example 3 – Condition for risk profile.

All three sub-conditions must be true before the whole condition is true, hence we are using **and** to separate the three sub-conditions.

The first sub-condition uses the **like**-operator to validate that the declared commodity code begins with the digits "1234".

The second sub-condition is a coordinate which returns the total sum of all gross weights declared on the item. If the total gross weight is more than 10kg the sub-condition is true; otherwise it is false. (This is similar to the cube shown in Figure 75)

The last sub-condition includes a test for each declared attachment. The test is divided into two parts, i.e. the certificate type must be "ABC" and the attachment must be required. If both of these parts are true the test returns 1; otherwise 0 is returned.

The `SUM( GsAttachmentItems )` tells that we adds the number of attachments which fulfils the criteria just determined. If none of the attachments fulfils the criteria the third condition is true; otherwise it is false. This is illustrated at Figure 84.

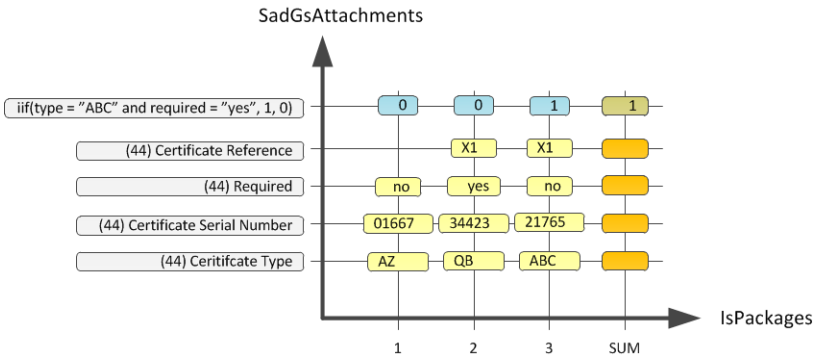


Figure 84: Counting the number of declared, required certificates having the type "ABC".

5.7.7.3.6 Syntax

In the previous section you have probably noticed that the some colons and dots are used when specifying the derived members and aggregations for the risk profile conditions. Figure 85 shows the basic format.

Figure 85: Basic format of a derived member.

When making a derived member the part before the double colon (::) defines the target cube. The target cube is the cube which will be extended by the new, derived member.

The subsequent parts are used for specifying the measure which is included in the calculations.

The part before the single colon (:) defines the name of the source cube, the part before the dot (.) specifies the dimension in the source cube (typically the measure dimension) and finally the part after the dot (.) specifies the name of the measure.

In most situations you do need to specify the complete qualified source, e.g. the name of the measure dimension shall always the same name as the cube. The system recognizes this, hence the specification can be reduced to the form shown in Figure 86.

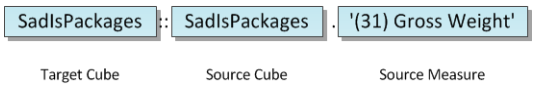


Figure 86: Reduced format of a derived member.

As seen in Figure 83 the target cube can be put in from of an equation rather than directly in front of a measure. This is used to indicate that the coordinate value that it represents is in the specific cube. This is shown in Figure 87.

Figure 87: The target cube is in front of an equation rather than directly in front of a measure.

5.7.7.4 Deleting, Undoing and Redoing Conditions

If you wish to delete part of the text in the expression box, position the curser where you wish to delete and then use the **Delete** key or the **Backspace** key on the keyboard. Alternatively you can highlight the area you wish to delete and then press the **Delete** key.

Should you find that you made a mistake in deleting an area you can easily bring it back using the **Undo** button, . You also have access to the **Redo** button, , and using this together with the **Undo** button enables you to toggle deleted conditions and texts.

☞ **Note:** As you delete, undo or redo conditions the same action takes place in the Partial Expression View if possible. If you for instance delete a condition it will automatically be deleted in the Formalized Expression View.

5.7.7.5 Validation of Rules

The content of the expression box is automatically validated to ensure that it complies with the syntax language in the system. If you have made a mistake an error message is displayed in the Validation View. In the example below one parenthesis – the last one – has been left out. Besides informing of an error the message also indicates the cause of the error by stating that a right parenthesis was expected. To correct the error you add the parenthesis and click the **Apply** button whereby the error message disappears.

Formalized Expression View
%1('(34a) Country of Origin' <> "CN")
and %2('(33.1) Commodity Code' = "0703200000"
Apply
ERROR : [4,1] Syntax error!: expecting

Figure 88: Example of error message

5.7.8 Indicators

In the Risk Profile Sub–tab there are two **indicators** to inform you of the status the formalization, see Figure 55above. The table below lists the possible options.

Indicator	Status
	(Blue) Indicates that the formalization is extended. This means that the formalized expression contains more information than what is included in the partial expression.
	Indicates that the formalization is valid.
	Indicates that the formalization is not valid.
	(Yellow) Indicates that there are warnings – i.e. there are things in the formalization that the user should pay attention to – e.g. that fields have been used that do not belong to all the selected declaration types.

5.7.9 Operators

The following table lists the operators that are defined in the system.

Symbol	Name	Logical value
=	equal to	*True*, if the left and right side are equal.
<>	different from	*True*, if the left and right side are different.
>	Greater than	*True*, if the left side is bigger then the right side. For text fields, *True* if the left side comes after the right side in an alphabetical comparison
<	Lesser than	*True*, if the left side is less then the right side. For text fields, *True* if the left side comes before the right side in an alphabetical comparison
>=	Greater than or equal to	*True*, if the left side is bigger then or equal to the right side. For text fields, *True* if the left side comes after the right side in an alphabetical comparison or if the two text fields are identical.
<=	Lesser than or equal to	*True*, if the left side is less then or equal to the right side. For text fields, *True* if the left side comes before the right side in an alphabetical comparison or if the two text fields are identical.
@	One of	*True*, if the left side is contained in the set specified on the right side.
!@	Not one of	*True*, if the left side is not contained in the set specified on the right side.
Like	Like	*True,* if the left side matches the pattern specified on the right side

5.8 Testing Risk Profiles

This section describes the Testbed function in ESKORT Clearance Workbench.

5.8.1 Background

Creating a Risk Profile can be a complex process especially if the Risk Profile is aimed to target risk which depends of complex combinations of the contents of the declaration which are being analyzed by the system.

After you have made a Risk Profile you are maybe unsure whether the Risk Profile actually is doing what is was supposed to. If the only way to determine this is by activating the Risk Profile and to wait for a declaration to be caught by the Risk Profile then it will be connected to discouragement and maybe frustrations if the Risk Profile does not fire as intended.

To help you avoiding this situation ESKORT Clearance Workbench provides a feature called a Testbed in which the Risk Profile can be tested against declarations before the Risk Profile is included in the risk analysis being performed on the real time declarations.

This section describes how the Testbed can be used to support you when creating Risk Profiles and to make you improve your skills by writing more complex Risk Profiles along with more confidence.

5.8.2 General Concept – Test Task

To verify that a Risk Profile is catching the expected declarations you need to specify a test of the Risk Profile – you do that by defining a Test Task along with one or more declarations as the Test Data Set for the Test Task.

When the Test Task is submitted all the declarations defined in the Test Data Set are analyzed by the Risk Profile in question and a Result is generated for each of the contained declarations. The Results are stored in the Test Task. Hereby a Test Task is constituted by:

- **Test Data Set**
  This is the declarations defined to be analyzed by the Risk Profile in question.
- **Analysis Results**
  This is the output from the Test Task, i.e. an Analysis Result is provided for each declaration defined to the Test Task.

When the Test Task has completed you can examine the Analysis Result generated by the Test Task.

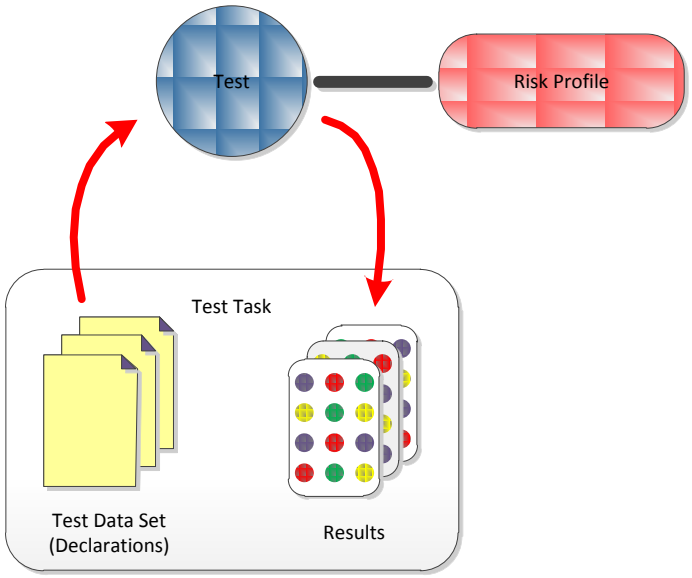


Figure 89: The Test Task contains declarations as Test Data, which are processed during the test. The Results is the output of the test and are stored along with the Test Task.

5.8.3 Results

The Testbed enables you to view the details of the Results generated when the Test Task was executed.

Firstly you can find the number of hits on each Declaration made by the Risk Profile.

Secondly you can examine the details of the Result provided, i.e. the output made by the Testbed Analysis Server. In the details you will find all the information by the Testbed Analysis Server provided through the configuration of the system.

5.8.4 Optimizing Risk Profiles

One the major benefits of using the Testbed is that that the Testbed enables you to change the formalization of the Risk Profile and almost immediately see which effects the change has caused. Hereby you are able to trim your Risk Profile so it finally catches the declarations which it was supposed to catch without catching a huge number of declarations which it was not supposed. Figure 90 illustrates the process of creating and improving a Risk Profile.

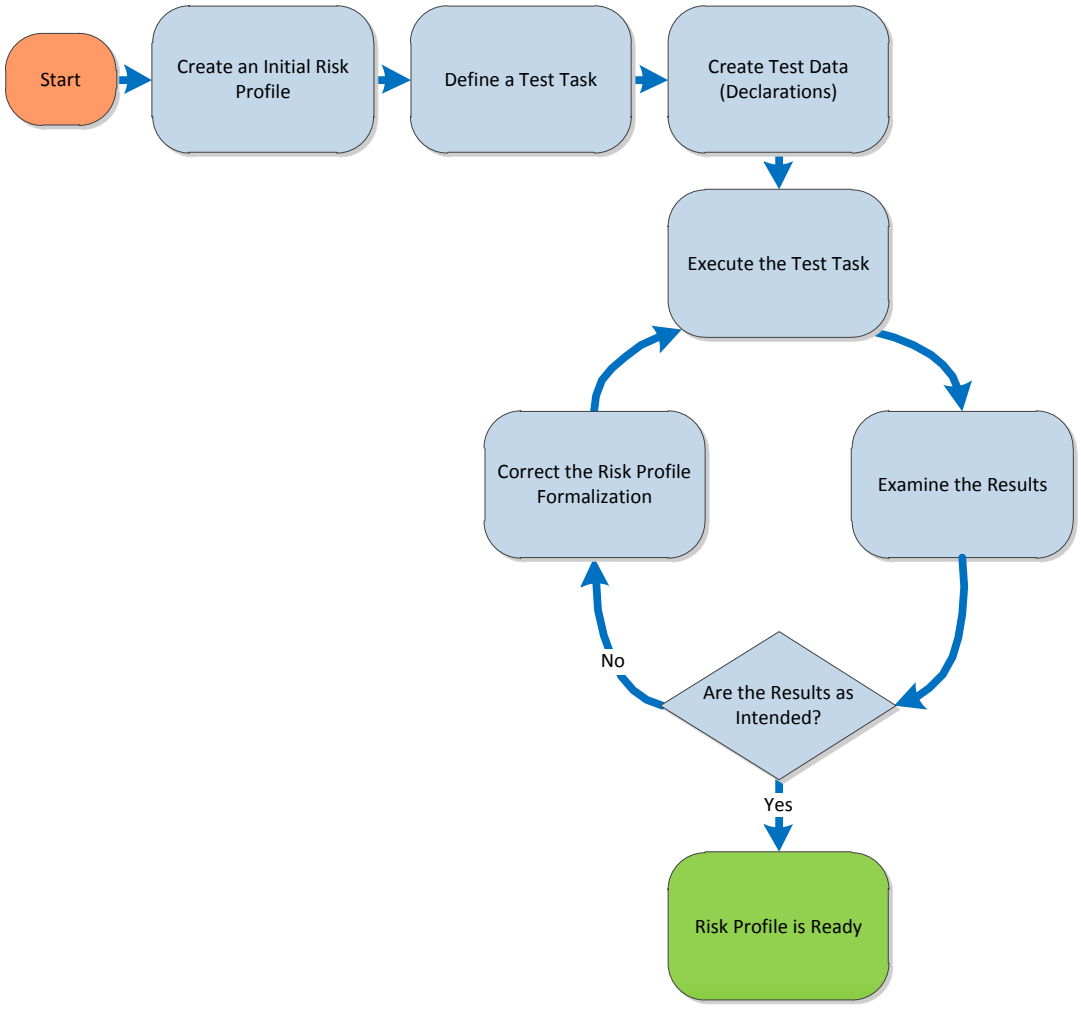


Figure 90: Process for optimizing the Risk Profile using the Testbed.

The process is described in the following sections.

5.8.4.1 Create an Initial Risk Profile

When you have found a method by which you think a risk can be identified you will typically formalize the method by creating a Risk Profile.

Create the Risk Profile and fill all the required information as well as the formalization part. Take your time and ensure that the descriptions, declaration types, etc. has been typed in as required. Do not make it active until you are sure that the Risk Profile works as intended. You should now save the Risk Profile.

You have now created the first revision of the Risk Profile; however to improve the performance of the Risk Profile you should now continue on the optimization process.

5.8.4.2 Define a Test Task

In the Detail-pane select the "Test Task"-tab and make a new Test Task. This test task will be the component through which the optimization process is done.

5.8.4.3 Create Test Data (Declarations)

Now you must add the Test Data to the Test Task. This can be done in different ways, i.e. by creating the declarations from scratch, importing Test Data from previous Test Tasks or from a common storage of Test Scenarios. Regardless of the origin of the Test Data it is important that the Test Data represents both declarations, which are supposed to be caught by the Risk Profile as well as declarations, which are not supposed to be caught.

If you are defining only declarations which are supposed to be caught you could easily end up having a Risk Profile which is to general, i.e. it is catching way too many declarations that were not intended to be caught – we say that the Risk Profile is too loose.

In the other hand, if you are defining too many declarations which are not to be selected compared to the number of declarations that should be selected, you could easily end up having a too narrow Risk Profile, i.e. a Risk Profile which only catches a reduced number of the declarations that were supposed to be caught – we say that the Risk Profile is too tight.

For that reason it is important that you find a good balance of declarations which are supported to be caught and declarations which are not.

5.8.4.4 Executing the Test Task

When the Test Data is in place you can execute the Test Task. The Test Task will apply the formalization of the Risk Profile on each of the declarations defined as the Test Data and will store the result from each declaration in the Test Task.

5.8.4.5 Examine the Results

When the Test Task has completed you can view the Results in the Test Task in which you will find an overview of how many hits each declaration has received by the Risk Profile.

By comparing the Result with the declarations defined when you created the Test Data you will be able to determine the quality of the Risk Profile.

Figure 91 illustrates how the Test Data can be illustrated along with the quality of the Risk Profile.

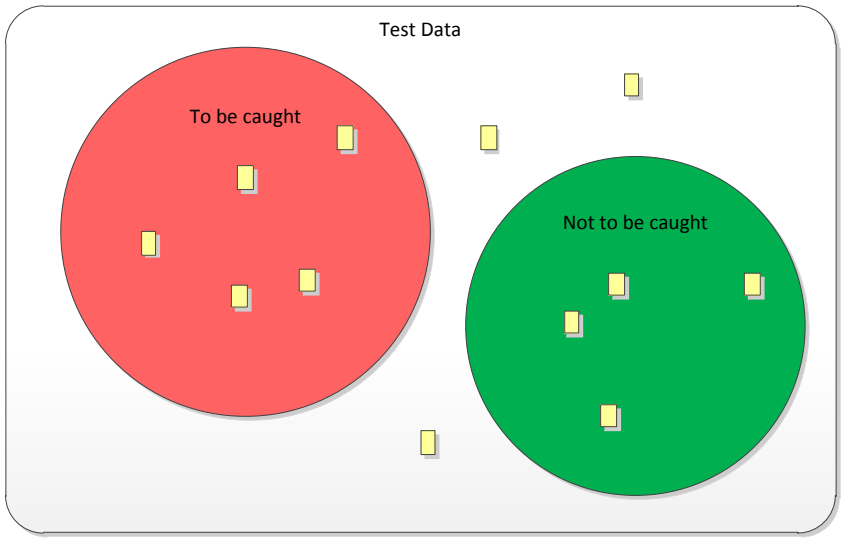


Figure 91: Test Data and their intension.

The rounded rectangle represents the set of all declarations (yellow rectangles) constituting the Test Data.

The red circle represents the set of the declarations which are supposed to be caught by the Risk Profile; whereas the green circle represents the declarations which are not supposed to be caught. The declarations located outside the red and the green circle are declarations for which in theory it does not really matter whether they are selected or not – these declarations represents noise in the model; however they can inform you about the trend of the Risk Profile.

Figure 92 shows the ideal Risk Profile catching only the declarations that were intended. This is illustrated by the blue circle with the bold border.

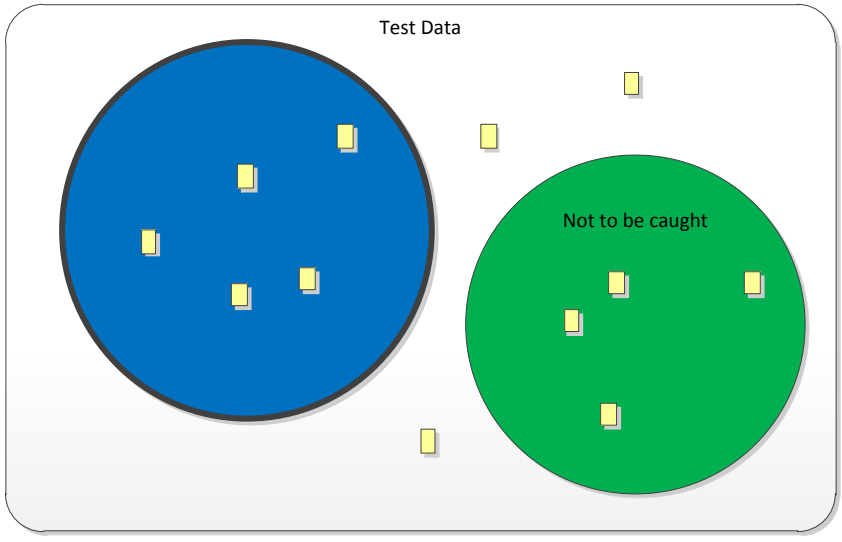


Figure 92: Ideal Risk Profile.

Writing a complex Risk Profile it can be hard (and maybe impossible) to create the ideal Risk Profile – often you will find a solution which is close to the ideal but not perfect – we call it "good". This is illustrated in Figure 93.

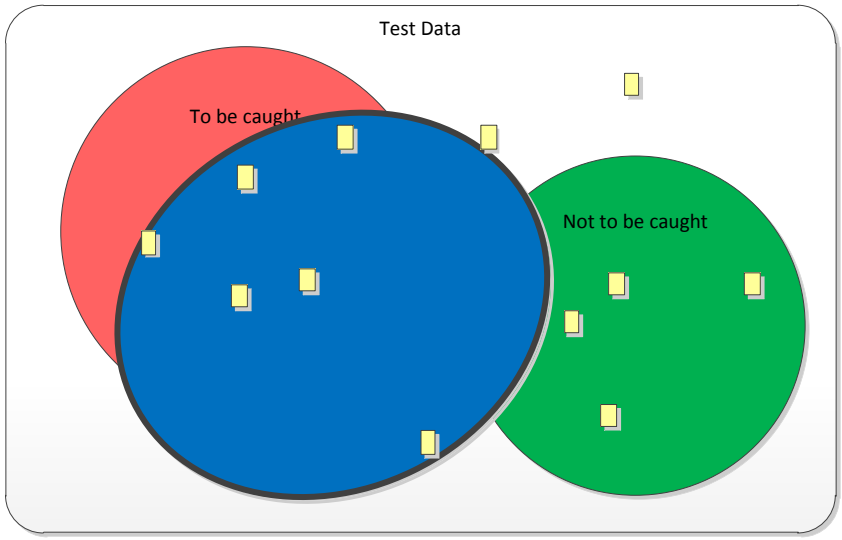


Figure 93: A good Risk Profile.

The figure shows a situation where almost all declarations from the red circle have been caught however a few other declarations has been caught as well, and we can expect that some of the declarations which were intended to be caught are not.

If you find that the Results comply with the intended distribution of hits you have verified the quality of the formalization part of the Risk Profile and you can activate the Risk Profile with confidence.

In the other hand, if you find a preponderance of declarations which have been caught that was not supposed to be caught you probably need to add more criteria to the formalization to tighten the Risk Profile avoiding the unintended declarations to be caught. See the Figure 94.

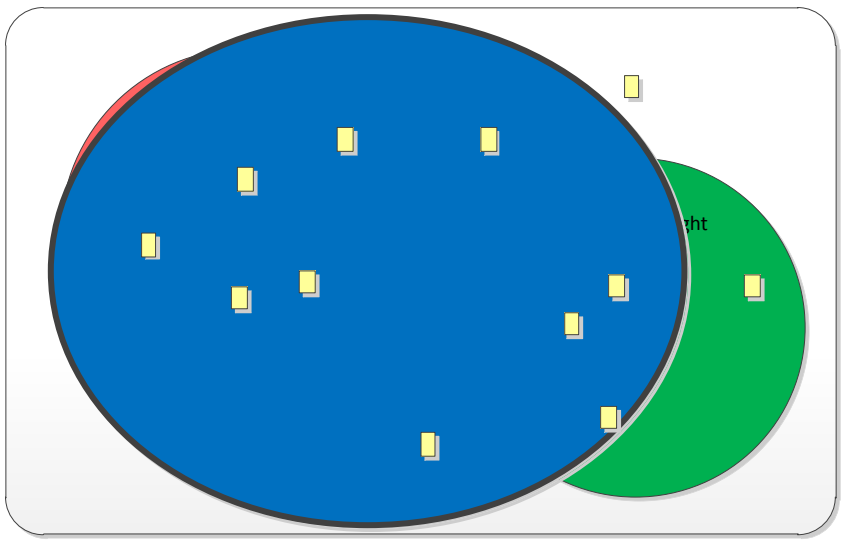


Figure 94: A too loose Risk Profile.

Opposite this if you find that too few of the declarations are caught, i.e. typically that some declarations which are supposed to be caught are not caught, then you need to loosen up the criteria so more of the declarations are caught. This situation is illustrated in Figure 95.

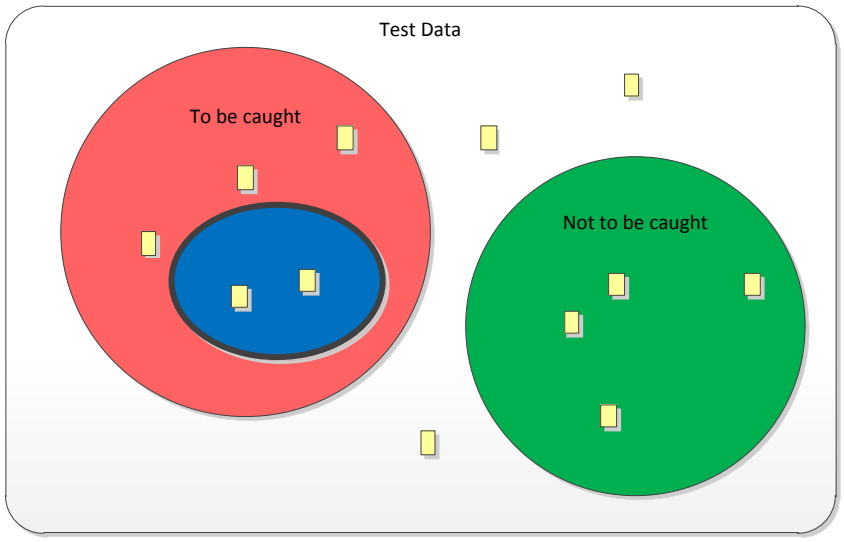


Figure 95: Too tight Risk Profile.

To determine which criteria to loosen up you should compare the contents of the declarations which intentionally are being caught with the contents of the declarations which are not caught, but which is was intended to be caught. The difference between the declarations can be a guide to find a way to loosen the criteria in the correct way.

After having changed the formalization you should re-execute the Test Task once more and subsequently re-examine the Results. By doing so you actually create a new Test Task in fact the original Test Task will be retained along with its Results.

5.8.5 Working in the Testbed

The purpose of the Testbed facility is to allow you to test risk profiles and establish that they will trap the desired declarations before they are put into production. In order to do such tests you define test tasks, execute these and monitor the result.

Test tasks can be defined in two ways – either integrated in risk profiles or as test scenarios which then can be used in any risk profile.

5.8.5.1 Test Tasks in Risk Profiles

When you work with test tasks in a risk profile you first must define the details for the risk profile and save it. Refer to section **Error! Reference source not found.** for details about how to define risk profiles. Also the risk profile must be active; otherwise you cannot set up test tasks and run tests.

In order to work with test tasks you select the Test Tasks sub-tab, see Figure 96 below.

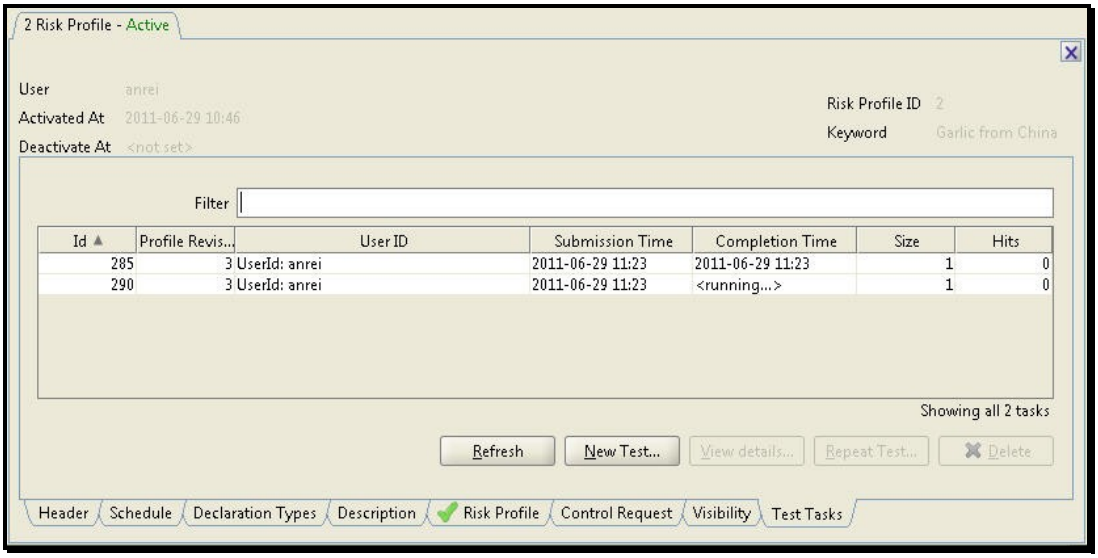


Figure 96: The Test Tasks sub-tab

The sub-tab displays a list of the test tasks that are defined and the relevant information about each of them such as submission and completion times and number of hits. The Filter box can be used to filter the list if so needed. At the bottom there are a number of buttons to be used for working in the sub-tab. The Refresh button can be used to refresh the list.

5.8.5.1.1 Define a New Test Task

In order to define a new test task you click the **New Test** button whereby the **Test Tasks Details** dialog is opened, refer to Figure 97 below.

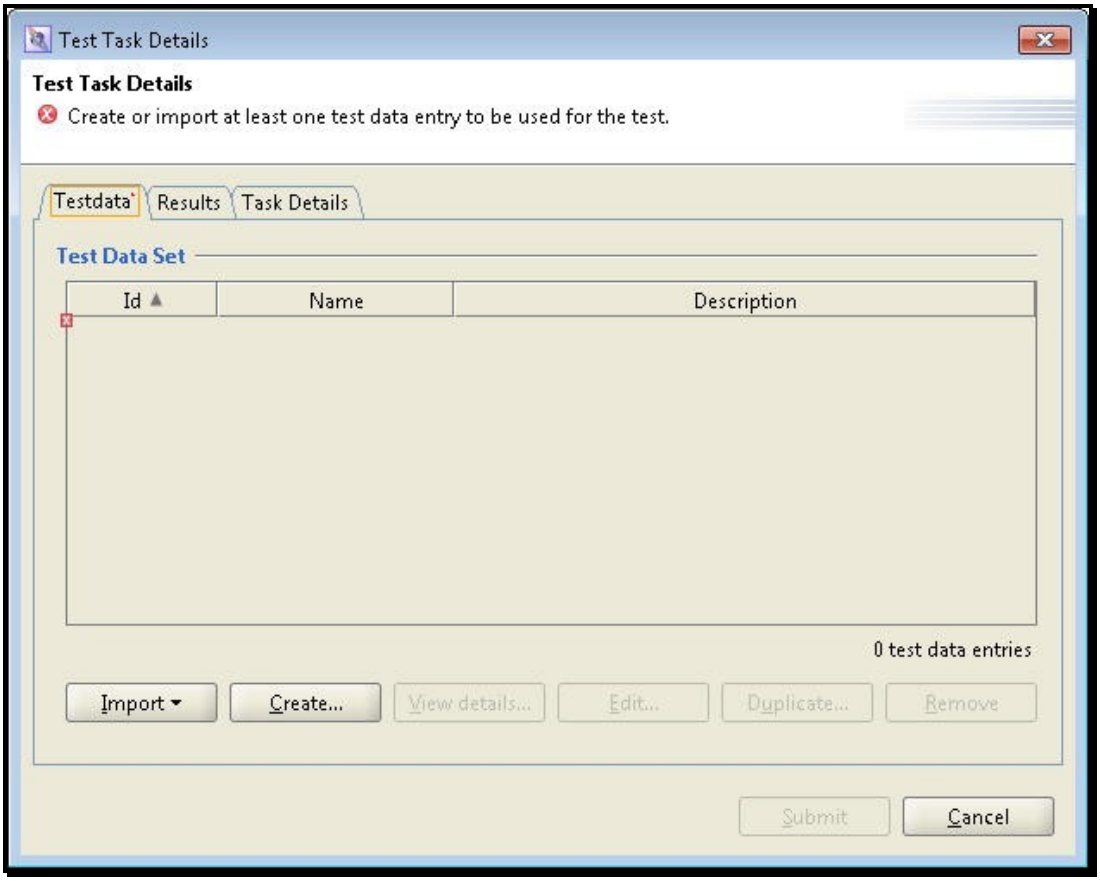


Figure 97: The Test Tasks Details dialog

The **Test Tasks Details** dialog is used for defining the test data, i.e. the declaration data, to be used for test. If you already have a test task that you previously have defined you can use the **Import** button to import the test task. You can either import from previous test tasks or from test scenarios. If you wish to define a new test data set you click the **Create** button and as result the **Test Data Properties** dialog is opened, see Figure 98 below.

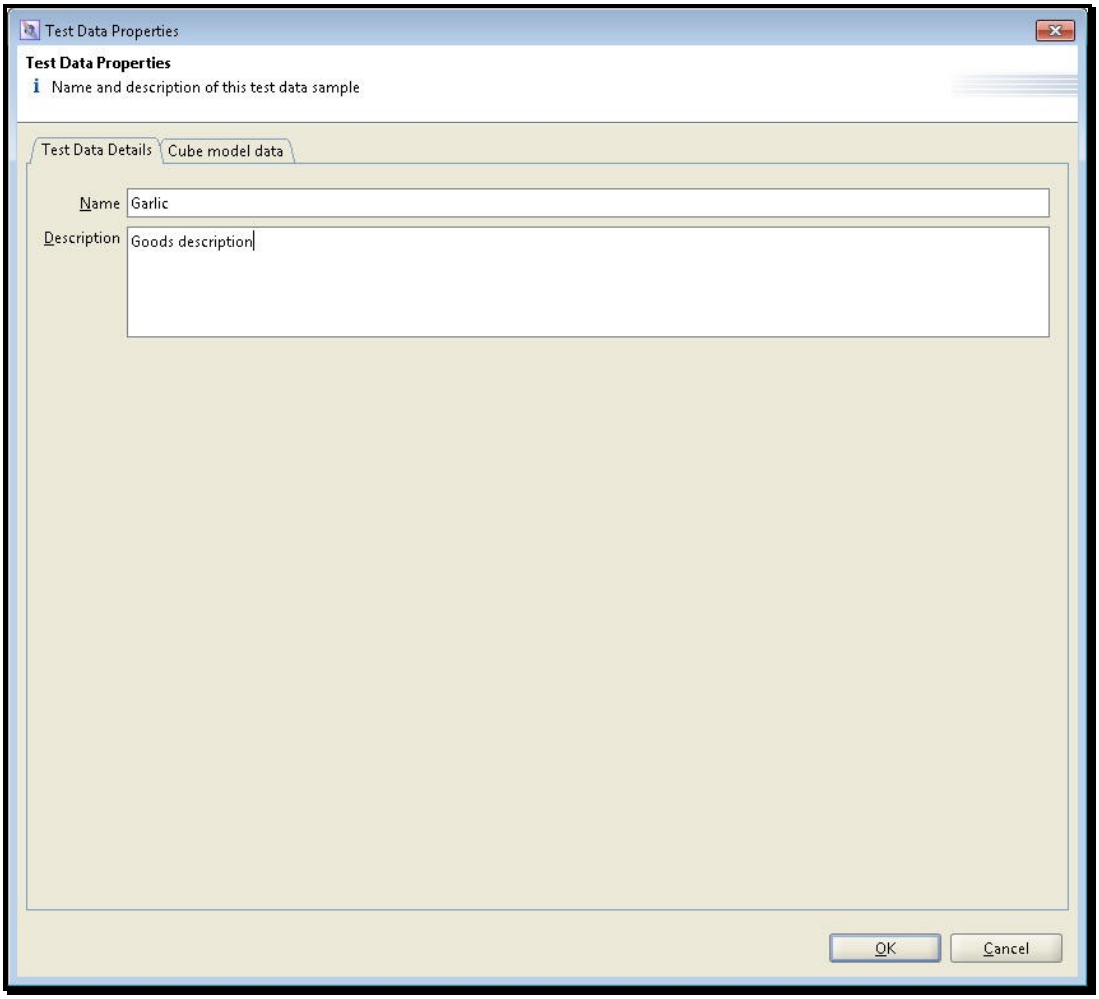


Figure 98: The Test Data Properties dialog

In the Test Data Details tab you must provide a name for the details which is used for later identification. The **Description** box can be used for writing a description but this is optional.

The Cube Model Data tab is used for defining the test data details, i.e. declaration data. The top part of the tab displays the cubes that are defined in the system and lower part displays the dimensions. If you e.g. select the Declaration cube the fields for the declaration header are displayed, see Figure 99Figure 99 below.

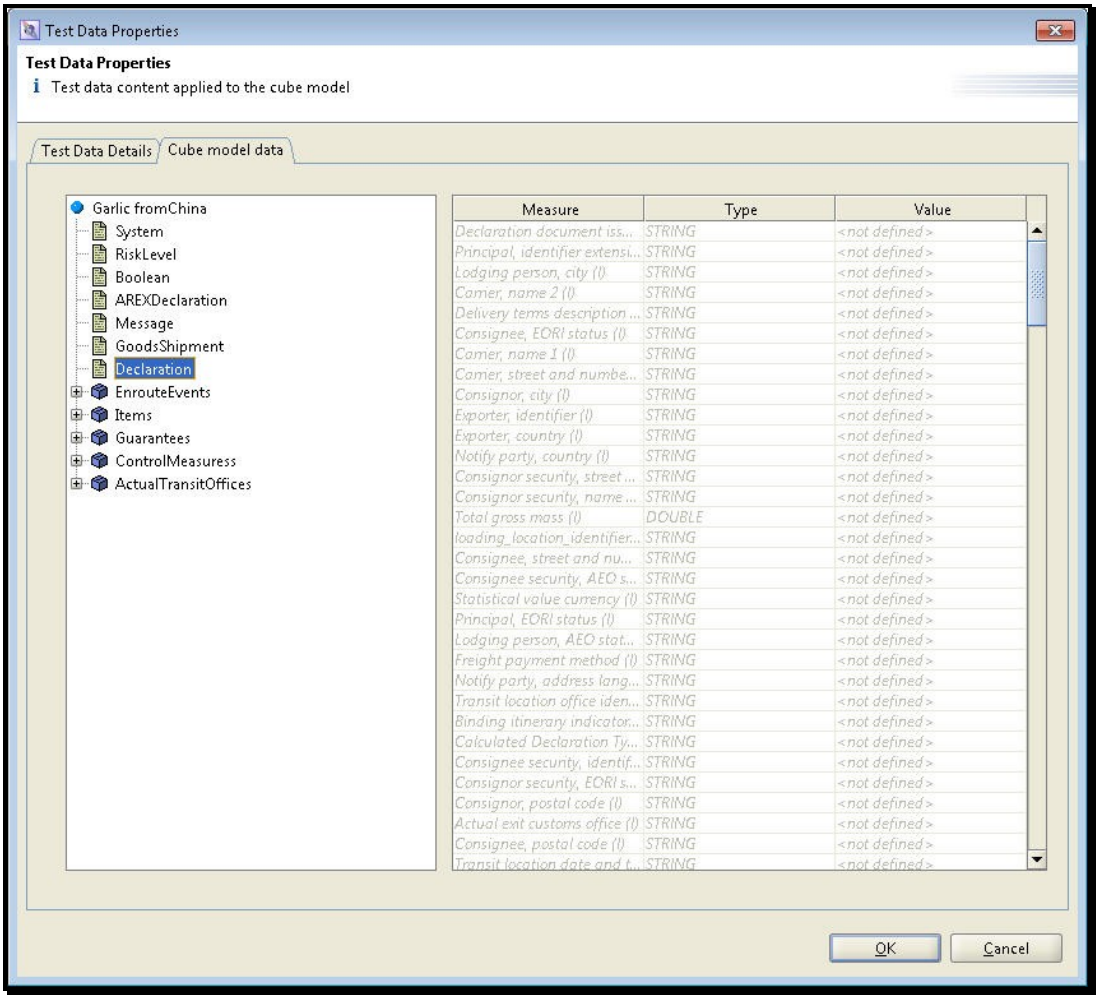


Figure 99: The Declaration header fields

You can now select a declaration field and key in the relevant string or value in the value field. However, for this declaration we will use the goods description field and this is located on the item level of the declaration. Thus, you click on the plus for the **Items** to expand it and then click on the **New Member** node. As a result the dimensions for the Item dimensions are listed – **Item** and **SadItemsEvaluation**. You now select the Item dimension whereby the declaration fields (or measures) for the item are displayed. In this test declaration the goods description "Garlic" will be used so you locate the **Goods Description** field and key in the code "Garlic" in the **Value** field, see Figure 100Figure 100 below.

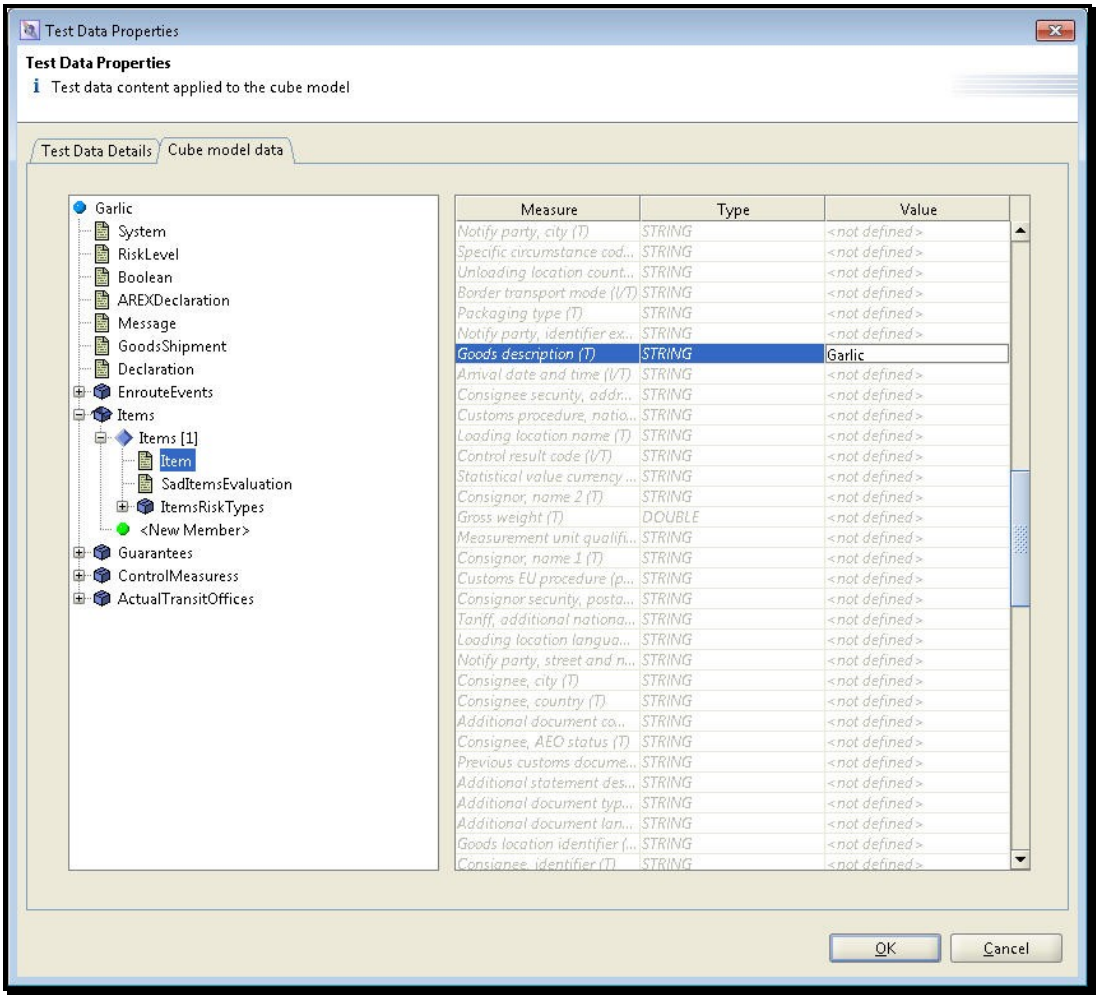


Figure 100: Defining goods description

Then you click OK to return to the **Test Tasks Details** dialog and the test data (declaration data) are defined. You can use the buttons to view the details, edit, duplicate or remove the test data. When finished you click the **Submit** button and the new test task is added in the Test Tasks sub-tab, see Figure 101 below.

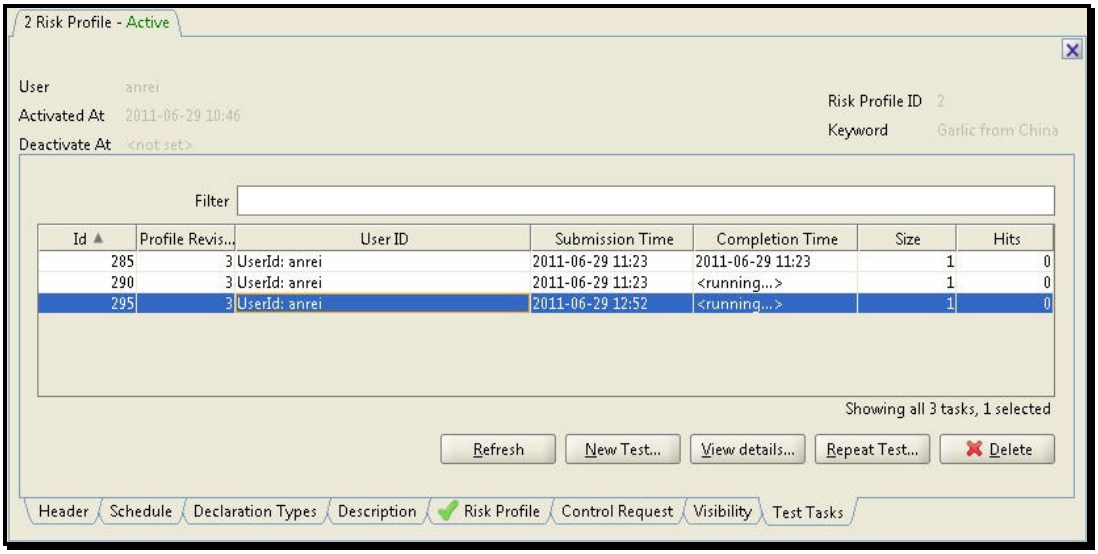


Figure 101: New test task added

You can use the buttons to view the details, repeat the test or delete the test task.

5.8.5.1.2 Test Results

When you submit a test task the **Completion Time** field in the Test Tasks sub-tab will display the text "running" to indicate that the test is executed. When the test is finished the completion time will be displayed. You can then study the result by clicking the **View details** button whereby the **Test Tasks Details** dialog is opened. In this dialog you select the **Result** tab, see Figure 102 below.

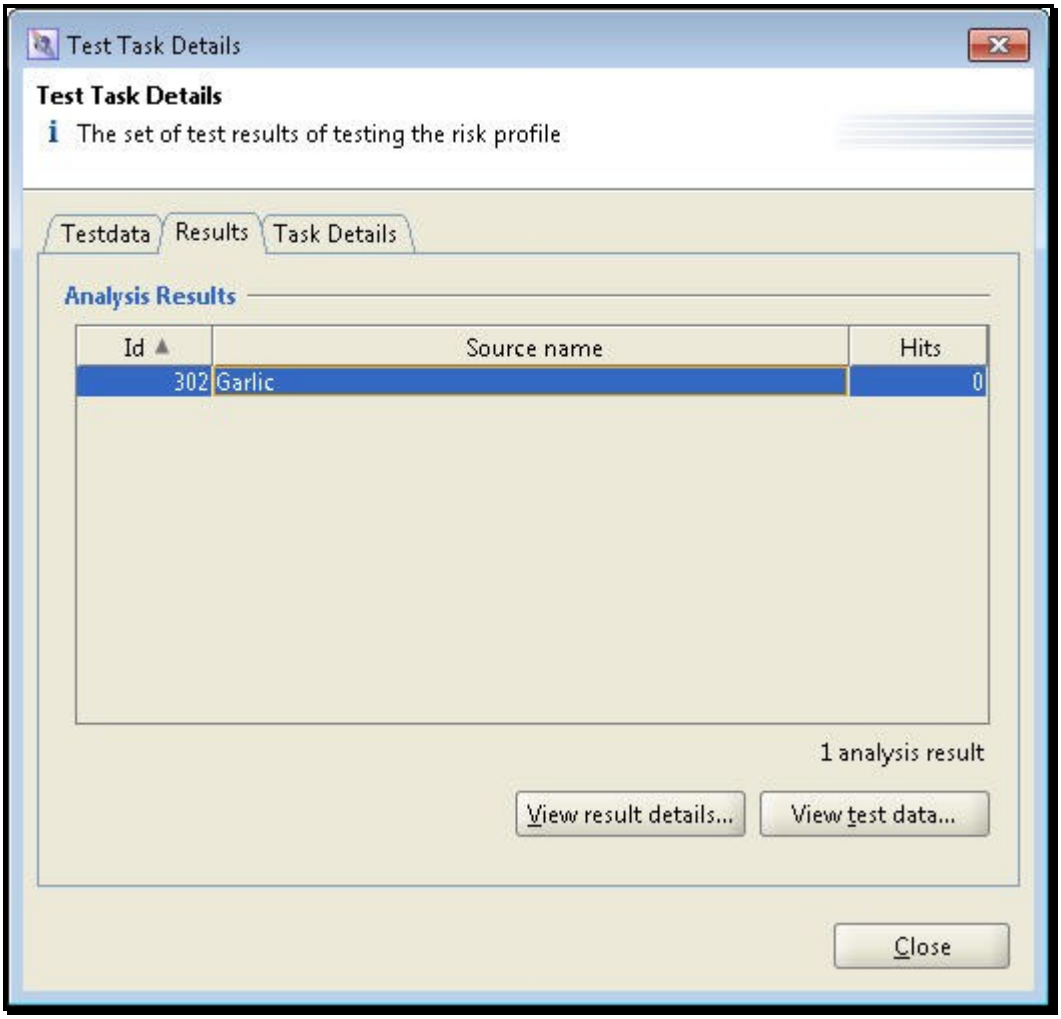


Figure 102: The Results tab

By clicking the **View result details** button you can now study the result of the test in the **Analysis Result Details** dialog.

5.8.5.2 Test Scenarios

In order to make it possible create generic test tasks that can be used in any risk profile you can define test scenarios. In order to define a test scenario you open the **Tools** menu and select the **Manage Test Scenarios** menu item. As a result the **Test Scenario Manager** dialog is opened, see Figure 103 below.

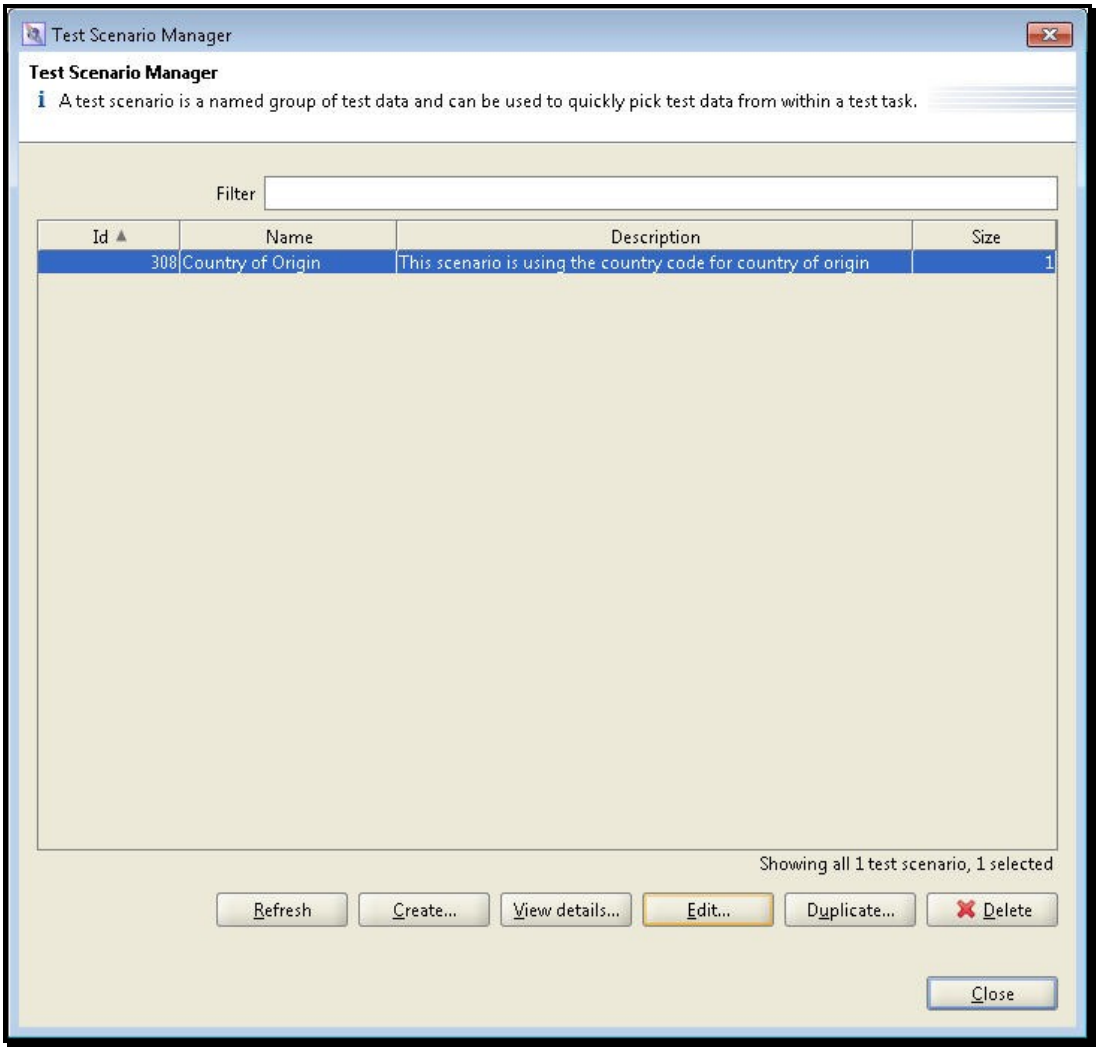


Figure 103: The Test Scenario Manager dialog

To define a new scenario you click the **Create** button to open the **Test Scenario Properties** dialog where you give the scenario a name and a description, see Figure 104 below.

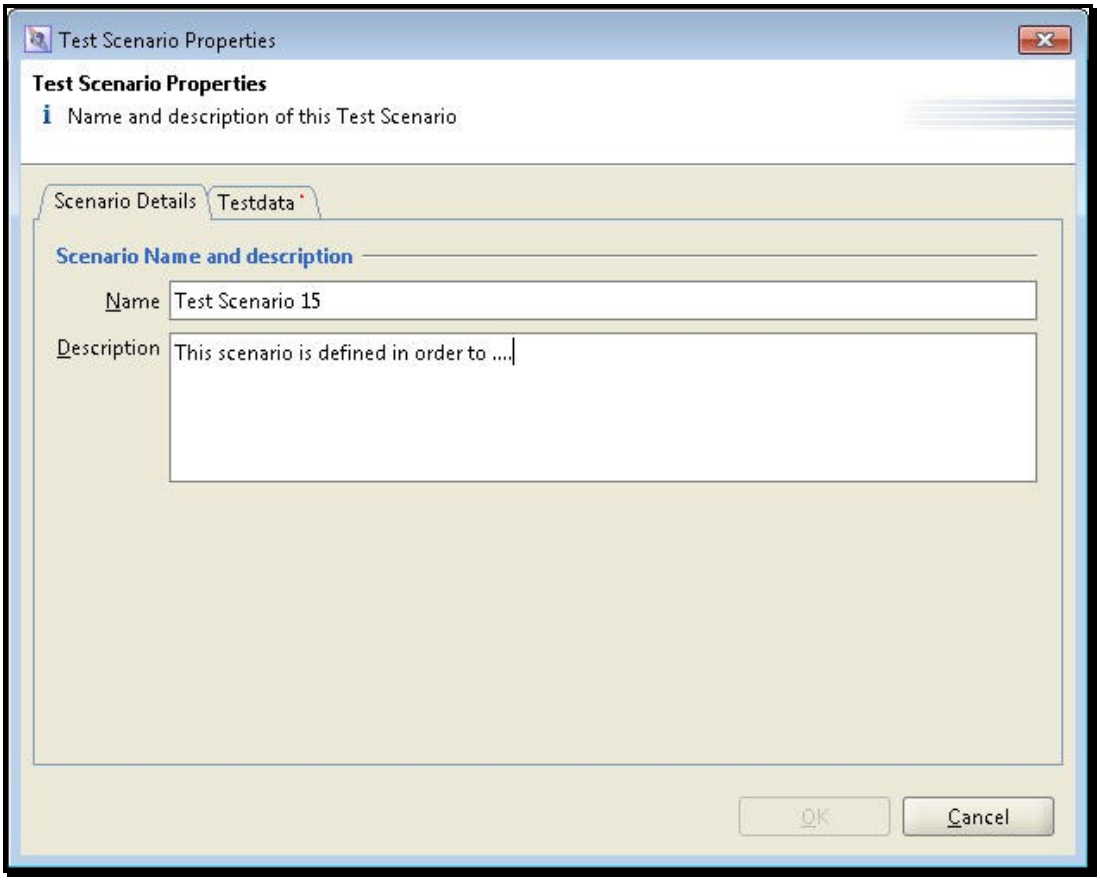


Figure 104: The Test Scenario Properties dialog

You then must define the test data so you select the **Testdata** tab and then click the **Create** button, see Figure 105 below.

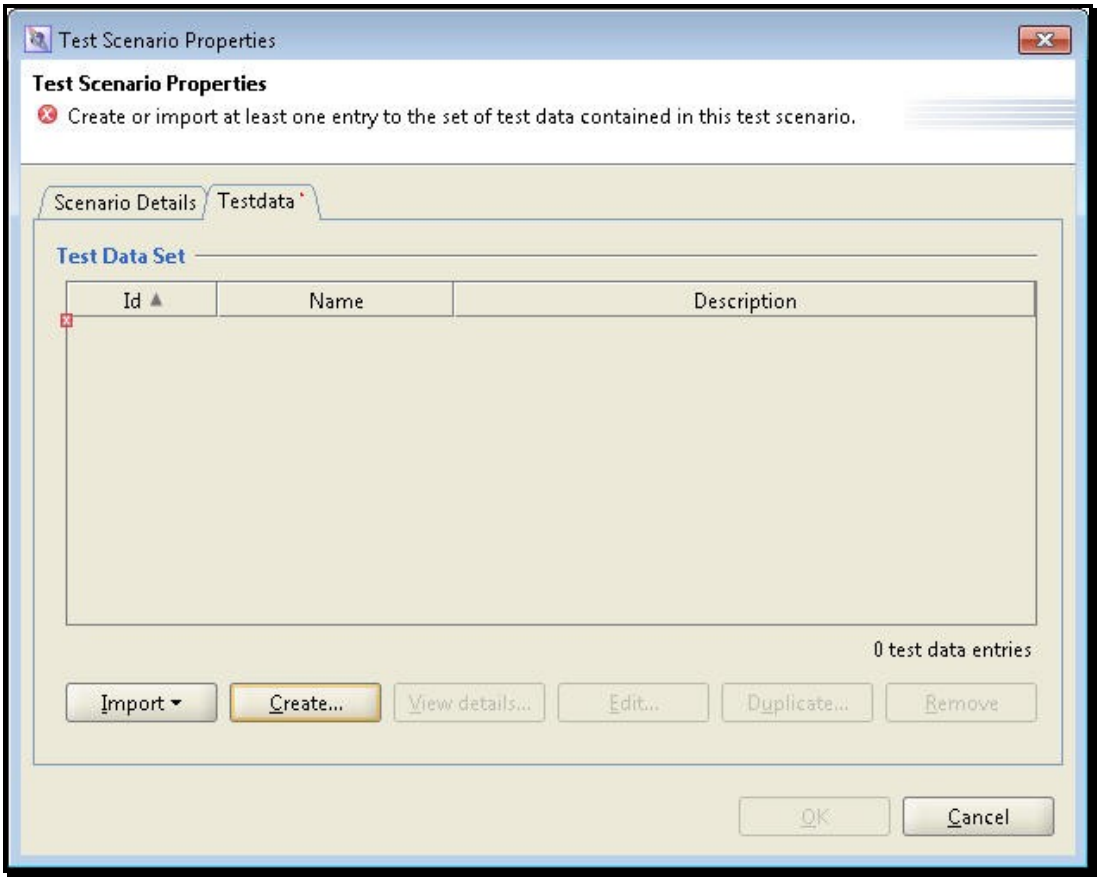


Figure 105: The Testdata dialog

For details of how to work with test data refer to section 5.8.5.1.1.

When you are finished defining the test data you click the OK and the new scenario is ready for use. You can use the buttons to view the details, edit, duplicate or delete test scenarios.

6. Working with the Compliant List

Working with the Compliant List you can add new traders to the list, edit existing traders, duplicate a trader in the list, delete traders from the list and make printouts. To work in the Compliant List, select the Compliant List tab.

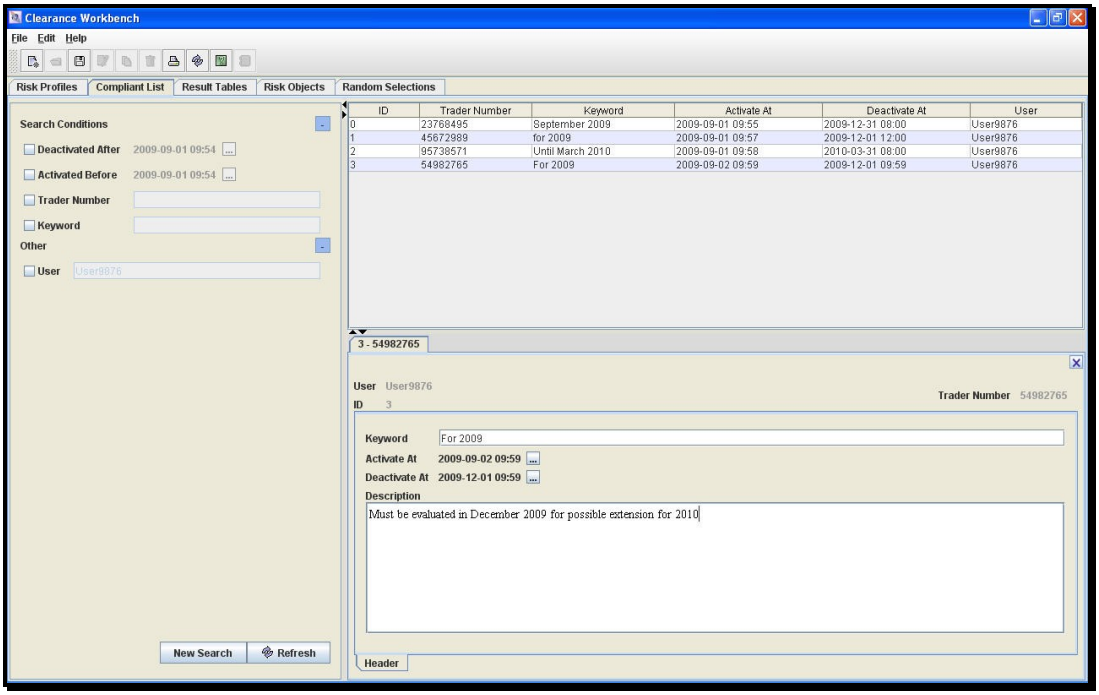


Figure 106: The Compliant List tab

6.1 Add a Trader to the Compliant List

To add a trader to the list, click the **New** button, , on the Toolbar. Alternatively, instead of clicking the **New** button you can select **New** from the File menu or use the shortcut key combination CTRL-N. As result, a new trader tab opens in the Data Area. In the **Trader Number** text box you enter the registration number of the trader and click the **Next** button. In the resulting detailed dialog for the trader you register the following:

- A Keyword as identification – for further information refer to section 4.5 Keyword
- Define Activation and Deactivation dates and times – for further information refer to section 4.3 Date and Time
- A textual description in the **Description** box as e.g. background information. Please note that entering a description is optional.

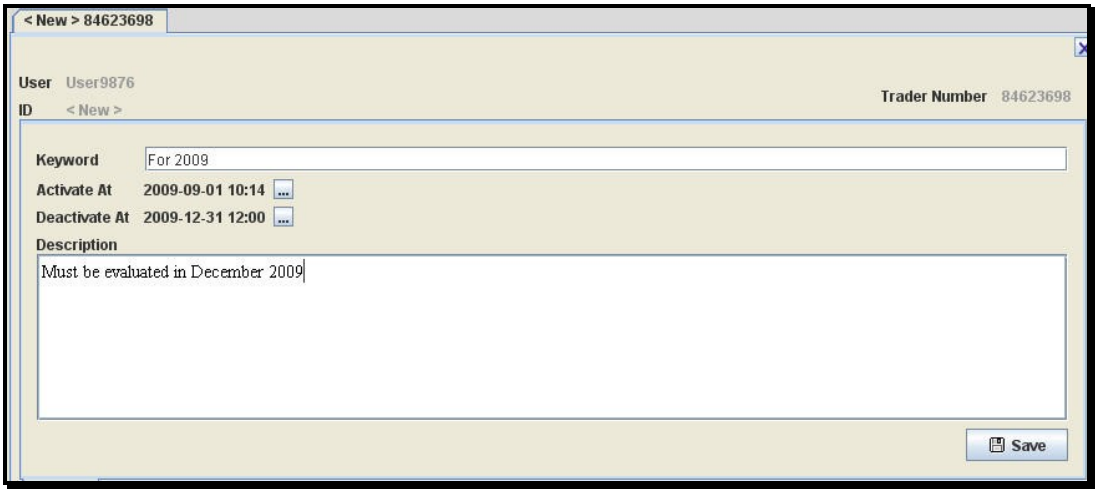


Figure 107: Details of a new Compliant List trader

You then click the **Save** button to save your work. Alternatively you can select the **Save** menu item from the **File** menu or use the shortcut keyboard combination CTRL-S.

6.2 Edit a Trader in the Compliant List

Locate the trader to be edited by using the filter criteria in the Filter Area. For further information on how to use the filtering criteria refer to

- Section 4.3 Date and Time for activation and deactivation times criteria
- Section 4.6 Trader Number for trader number criteria
- Section 4.5 Keyword for keyword criteria
- Section 4.7 User for user criteria

When you are working in the Filter Area it is possible to use wildcards in the text box to make easier to indicate a user for which you do not know the full code. For instructions about using wildcards refer to section 4.9 Wildcards.

☞ **Tip:** Instead of ticking the tick box you can click in the text box and as a result the tick box is ticked and the text box is opened for edit.

When you have located the trader, select the trader in the Table Area and click the **Open** button. Alternatively you can select the **Open** menu item from the **File** menu or use the shortcut keyboard combination CTRL-O. The trader is now open in read-only mode and to be able to edit it you click the **Edit** button. Alternatively you can select the **Edit** menu item in the **Edit** menu or use the shortcut key combination CTRL-E.

Now edit the details of the trader and for further information refer section 6.1 Add a trader to the Compliant List.

When you are finished editing the trader click the **Save** button to save your work. Alternatively you can select the **Save** menu item from the **File** menu or use the shortcut keyboard combination CTRL-S.

6.3 Delete a Trader from the Compliant List

Locate the trader to be deleted by using the filter criteria in the Filter Area. For further information on how to use the filtering criteria refer to

- Section 4.3 Date and Time for activation and deactivation times criteria
- Section 4.6 Trader Number for trader number criteria
- Section 4.5 Keyword for keyword criteria

If you wish to filter using the **User** criteria you tick the **User** tick box whereby the text box is opened and the cursor positioned to the left in the box. You then enter the user ID and click the **Refresh** button to activate the filter. Alternatively you can select the **Refresh** menu item from the **Edit** menu or click the **Refresh** button on the Toolbar.

When you are working in the Filter Area it is possible to use wildcards in the text box to make easier to indicate a user for which you do not know the full code. For instructions about using wildcards refer to section 4.9 Wildcards.

☞ **Tip:** Instead of ticking the tick box you can click in the text box and as a result the tick box is ticked and the text box is opened for edit.

When you have located the trader, select the trader and then click the **Delete** button on the Toolbar. Alternatively you can select the **Delete** menu item from the **Edit** menu or use the **Delete** key.

6.4 Duplicate a Trader in the Compliant List

Instead of defining a new trader in the Compliant List it can sometimes be easier to duplicate an existing trader, make the necessary changes and then save it. For further information on how to duplicate refer to section 3.4.2.2 Duplicating Configuration Elements.

6.5 Printing

Printing from the Compliant List tab can be done in two different printouts – you can print the list of traders in the Table Area or you can print trader details in the Data Area. For information about printing in the Compliant List tab refer to section 3.4.2.4 Printing Configuration Elements.

7. Working with Result Tables

The main work in the Result Tables tab is to edit the threshold values that govern the classification into risk groups. Besides this you can also create new result tables, delete existing, duplicate existing and make printouts.

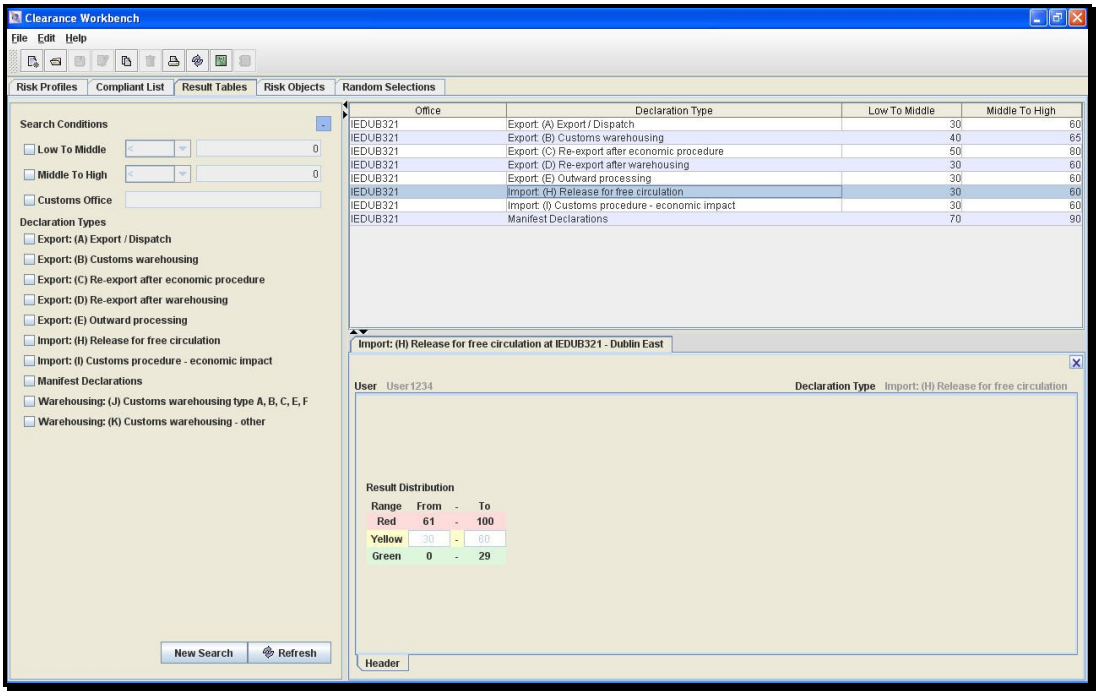

Clearance Workbench
File Edit Help
Risk Profiles
Compliant List
Result Tables
Risk Objects
Random Selections
Search Conditions
Low To Middle
Middle To High
Customs Office
Declaration Types
Export: (A) Export / Dispatch
Export: (B) Customs warehousing
Export: (C) Re-export after economic procedure
Export: (D) Re-export after warehousing
Export: (E) Outward processing
Import: (H) Release for free circulation
Import: (I) Customs procedure - economic impact
Manifest Declarations
Warehousing: (J) Customs warehousing type A, B, C, E, F
Warehousing: (K) Customs warehousing - other
New Search
Refresh
Office
Declaration Type
Low To Middle
Middle To High
IEDUB321 Export: (A) Export / Dispatch 30 60
IEDUB321 Export: (B) Customs warehousing 40 65
IEDUB321 Export: (C) Re-export after economic procedure 50 80
IEDUB321 Export: (D) Re-export after warehousing 30 60
IEDUB321 Export: (E) Outward processing 30 60
IEDUB321 Import: (H) Release for free circulation 30 60
IEDUB321 Import: (I) Customs procedure - economic impact 30 60
IEDUB321 Manifest Declarations 70 90
Import: (H) Release for free circulation at IEDUB321 - Dublin East
User User1234
Declaration Type Import: (H) Release for free circulation
Result Distribution
Range From - To
Red 61 - 100
Yellow 30 - 60
Green 0 - 29
Header

Figure 108: The Result Tables tab

7.1 Create a new Result Table

To add a new result table, click the **New** button, , on the Toolbar. Alternatively, instead of clicking the **New** button you can select **New** from the File menu or use the shortcut key combination **CTRL-N**. As result, a new result table tab opens in the Data Area. In the Declaration Type drop down box select the declaration type for which you wish to create a new result table and click the **Next** button. In the following dialog enter the threshold values for the orange group and click the **Save** button. Alternatively you can select the **Save** menu item from the **File** menu or use the shortcut keyboard combination **CTRL-S**.

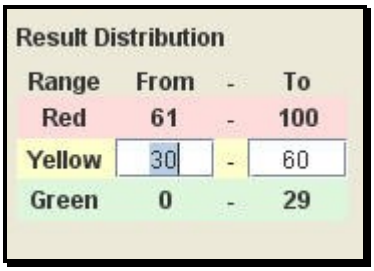


Figure 109: Changing threshold values for a result table

Figure 110: Setting threshold values for a new Result Table

7.2 Editing a Result Table

Locate the result table to be edited by using the filter criteria in the Filter Area. For further information on how to use the filtering criteria refer to

- section 4.4 Customs Office for customs office criteria
- section 4.1 Declaration Types for declaration types criteria

If you wish to filter using the **Low to Middle** or **Middle to High** filter criteria boxes you tick the tick box(es), select an operator from the **Operator** drop down box(es) and enter the threshold values in the text box(es).

☞ **Tip:** Instead of ticking the tick box you can click in the text box and as a result the tick box is ticked and the text box is opened for edit.

When you have located the result table, select it in the Table Area and then click the **Open** button. Alternatively you can select the **Open** menu item from the **File** menu or use the shortcut keyboard combination CTRL-O. The result table is now open in read-only mode and to be able to edit it click the **Edit** button. Alternatively you can select the **Edit** menu item in the **Edit** menu or use the shortcut key combination CTRL-E.

Now edit the threshold values and when you are finished editing the result table click the **Save** button to save your work. Alternatively you can select the **Save** menu item from the **File** menu or use the shortcut keyboard combination CTRL-S.

7.3 Deleting a Result Table

Locate the result table to be deleted by using the filter criteria in the Filter Area. For further information on how to use the filtering criteria refer to

- section 4.4 Customs Office for customs office criteria
- section 4.1 Declaration Types for declaration types criteria

If you wish to filter using the **Low to Middle** or **Middle to High** filter criteria boxes you tick the tick box(es), select an operator from the **Operator** drop down box(es) and enter the threshold values in the text box(es).

When you have located the result table, select it and then click the **Delete** button, , on the Toolbar. Alternatively you can select the **Delete** menu item from the **Edit** menu or use the **Delete** key.

7.4 Duplicate Result Tables

Instead of defining a new result table it can sometimes be easier to duplicate an existing result table, make the necessary changes and then save it. For further information on how to duplicate refer to section 3.4.2.2 Duplicating Configuration Elements.

7.5 Printing

Printing from the Result Tables tab can be done in two different printouts – you can print the list of result tables in the Table Area or you can print result table details in the Data Area. For information about printing in the Result Table tab refer to section 3.4.2.4 Printing Configuration Elements.

8. Working with Risk Objects

The work with risk objects consists of creating new risk objects, editing, duplicating or deleting existing risk objects and printing risk object details.

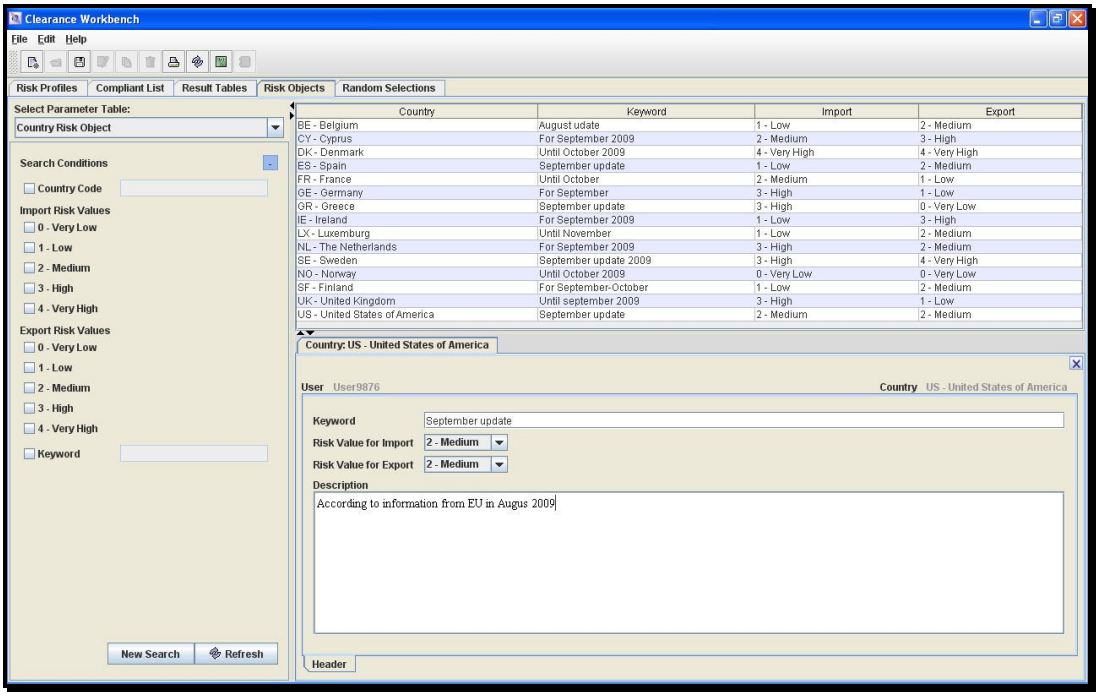


Figure 111: The Risk Object tab

8.1 Creating new Risk Objects

When you create a new risk object the work is executed in two slightly different ways depending on the Risk Object Type. Creating new Procedure, Trader and Commodity risk objects are all done in the same way while Country risk objects are created in a different way.

8.1.1 Creating new Procedure, Trader and Commodity Risk Objects

To add a new risk object, you first have to select the Risk Object Type from the **Select Parameter Table** drop down box and then you click the **New** button, , on the Toolbar. Alternatively, instead of clicking the **New** button you can select **New** from the File menu or use the shortcut key combination **CTRL-N**. As a result a new Risk Object tab opens in the Data Area. The tab contains a text box for entering a code and depending on what risk object type you are creating, the code can either be Procedure Code, Trader Code or Commodity code. You enter the relevant code in the text box and click the **Next** button.

In the dialog you then enter a keyword in the **Keyword** text box, for further information refer to section 4.5 Keyword, and a textual description in the **Description** text box e.g. the background for creating the new risk object. In order to assign a risk value to the risk object open the **Risk Value** drop down box, select a value and click the **Save** button, .

Alternatively you can select the **Save** menu item from the **File** menu or use the shortcut keyboard combination **CTRL-S**.

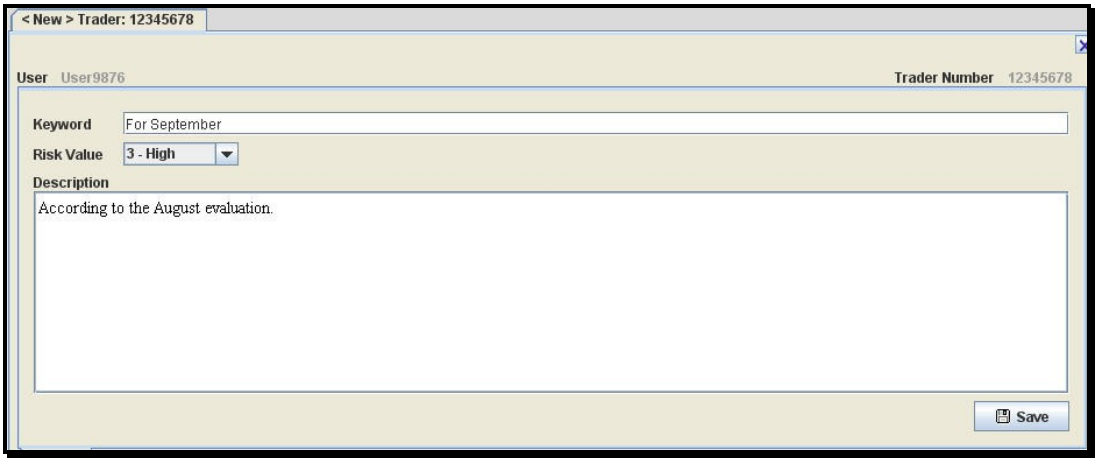


Figure 112: New trader risk object

8.1.2 Creating a new Country Risk Object

To add a new risk object you first have open the **Select Parameter Table** drop down box and select Country Risk Objects. You then click the **New** button, , on the Toolbar. Alternatively, instead of clicking the **New** button you can select **New** from the File menu or use the shortcut key combination **CTRL-N**. As a result a new Country Risk Object tab opens in the Data Area. The tab contains a Country drop down box for selecting the country code. Select the relevant country click the **Next** button.

The ***Clearance Workbench*** now tries to validate the code to establish if it is an existing, valid code. If the entered code is not valid an error message will be displayed and you have to correct the code. If the code is valid the Risk Objects detail tab is opened.

In the dialog you then enter a keyword in the **Keyword** text box, for further information refer to section 4.5 Keyword, and a textual description in the **Description** text box e.g. the background for creating the new risk object. For country codes you need to assign separate risk values for export and import. First open the **Risk Value for Export** drop down box and select a value and then repeat the process in the **Risk Value for Import** drop down box. Click the **Save** button, , on the Toolbar to save your work. Alternatively you can select the **Save** menu item from the **File** menu or use the shortcut keyboard combination **CTRL-S**.

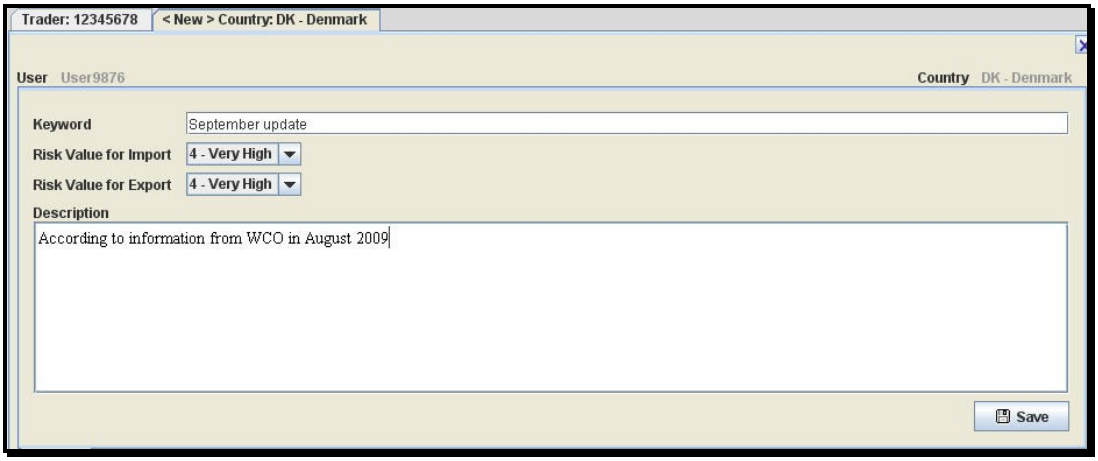


Figure 113: Setting risk value for country risk object

8.2 Editing a Risk Object

Locate the risk object to be edited by using the filter criteria in the Filter Area. The filter criteria consist of the **Risk Object Code**, **Risk Values** and **Keyword**. For country risk object the risk value criteria is separated for export and import.

If you wish to filter using the risk object code filter criteria you tick the **Risk Object Code** tick box and enter the code in the text box.

☞ **Tip:** Instead of ticking the tick box you can click in the text box and as a result the tick box is ticked and the text box is opened for edit.

In order to use the **Risk Values** tick boxes for filtering, tick the box or the boxes you wish to include in the filter criteria.

For information on how to use the **Keyword** filter criteria refer to section 4.5 Keyword.

If necessary you can use different combinations of filter criteria.

When you have located the relevant risk object, select it in the Table Area and then click the **Open** button, , on the Toolbar. Alternatively you can select the **Open** menu item from the **File** menu or use the shortcut keyboard combination **CTRL-O**. The risk object is now opened in read-only mode and to be able to edit it click the **Edit** button, , on the Toolbar. Alternatively you can select the **Edit** menu item in the **Edit** menu or use the shortcut key combination **CTRL-E**.

Edit the risk value(s), enter a keyword and a description. When finished, click the **Save** button to save your work. Alternatively you can select the **Save** menu item from the **File** menu or use the shortcut keyboard combination **CTRL-S**.

8.3 Deleting a Risk Object

Locate the risk object to be deleted by using the filter criteria in the Filter Area. The filter criteria consist of the **Risk Object Code**, **Risk Values** and **Keyword**. For country risk objects the risk values criteria is separated for export and import.

If you wish to filter using the risk object code filter criteria, you tick the **Risk Object Code** tick box and enter the code in the text box.

☞ **Tip:** Instead of ticking the tick box you can click in the text box and as a result the tick box is ticked and the text box is opened for edit.

In order to use the **Risk Values** tick boxes for filtering, tick the box or the boxes you wish to include in the filter criteria.

For information on how to use the **Keyword** filter criteria refer to section 4.5 Keyword.

If necessary you can use different combinations of filter criteria.

When you have located the relevant risk object, select it in the Table Area and then click the **Delete** button, , on the Toolbar. Alternatively you can select the **Delete** menu item from the **Edit** menu or use the **Delete** key.

8.4 Duplicating Risk Objects

Instead of defining a new risk object it can sometimes be easier to duplicate an existing risk object, make the necessary changes and then save it. For further information on how to duplicate refer to section 3.4.2.2 Duplicating Configuration Elements.

8.5 Printing

Printing from the Risk Objects tab can be done in two different printouts – you can print the list of risk objects in the Table Area or you can print risk objects details in the Data Area. For information about printing in the Risk Objects tab refer to section 3.4.2.4 Printing Configuration Elements.

☞ **Tip:** List of risk objects can be very substantial especially for commodity codes and trader number. Before printing such list you most probably need to filter the lists in order to only print the relevant risk objects.

9. Working with Random Selections

In the Random Selection tab you can create new Random Selection configuration elements, edit, duplicate and delete existing ones and make printouts.

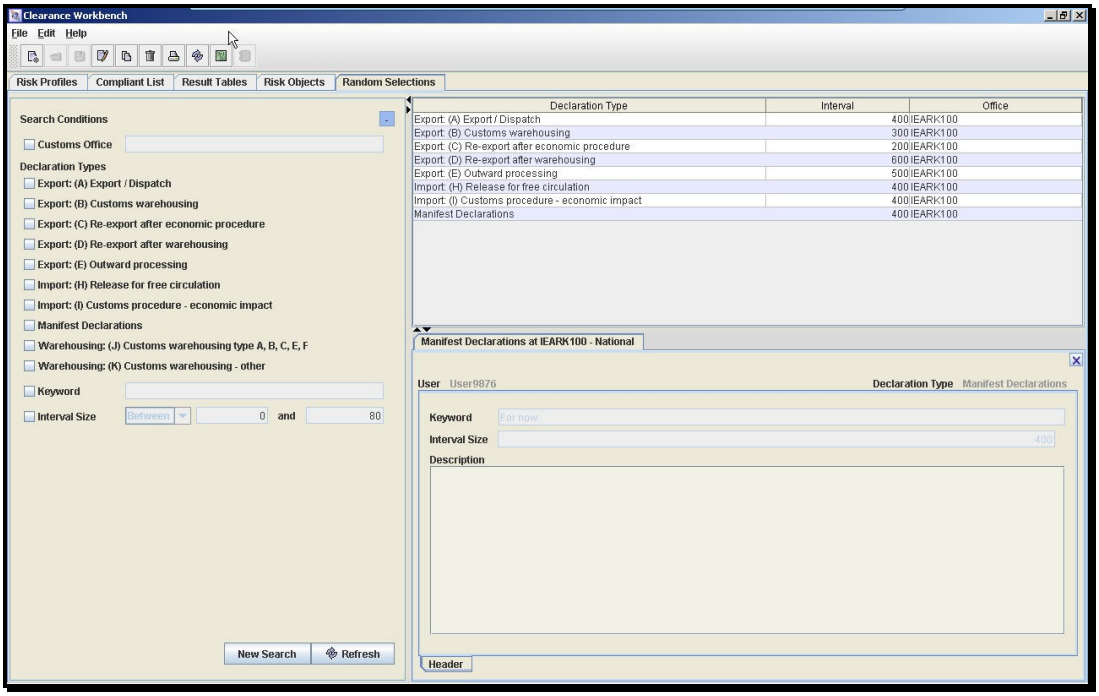


Figure 114: The Random Selection tab

9.1 Create new Random Selection Configuration Elements

To add a new random selection configuration element, click the **New** button, , on the Toolbar. Alternatively, instead of clicking the **New** button you can select **New** from the File menu or use the shortcut key combination **CTRL-N**. As result, a new random selection element opens in the Data Area. In the **Declaration Type** drop down box select the declaration type for which you wish to create the new random selection element and click the **Next** button. In the following dialog enter a keyword - for further information refer to section 4.5 Keyword and the interval size**.** The interval size cannot be below 99 and not more than 9999. Then write a description in the **Description** text box and click the **Save** button. Alternatively you can select the **Save** menu item from the **File** menu or use the shortcut keyboard combination **CTRL-S**.

9.2 Edit Random Selection Configuration Elements

Locate the Random Selection element to be edited by using the filter criteria in the Filter Area. For further information on how to use the filtering criteria refer to

- Section 4.4 Customs Office for customs office criteria
- Section 4.1 Declaration Types for declaration types criteria
- Section 4.5 Keyword for keyword criteria

If you wish to filter using the **Interval Size** filter criteria boxes you tick the **Interval Size** tick box, select an operator from the **Operator** drop down box and enter the threshold values in the text boxes.

☞ **Tip:** Instead of ticking the tick box you can click in the text box and as a result the tick box is ticked and the text box is opened for edit.

When you have located the random selection element, select it in the Table Area and then click the **Open** button. Alternatively you can select the **Open** menu item from the **File** menu or use the shortcut keyboard combination **CTRL-O**. The random selection element is now open in read-only mode and to be able to edit it you click the **Edit** button. Alternatively you can select the **Edit** menu item in the **Edit** menu or use the shortcut key combination **CTRL-E**.

In the dialog enter a keyword and the interval size**;** please note that the size cannot be below 99 and not more than 9999. Then write a description in the **Description** text box and when you are finished editing the random selection element click the **Save** button to save your work. Alternatively you can select the **Save** menu item from the **File** menu or use the shortcut keyboard combination **CTRL-S**.

9.3 Deleting Random Selection Configuration Elements

Locate the Random Selection element to be deleted by using the filter criteria in the Filter Area. For further information on how to use the filtering criteria refer to

- Section 4.4 Customs Office for customs office criteria
- Section 4.1 Declaration Types for declaration types criteria
- Section 4.5 Keyword for keyword criteria

If you wish to filter using the **Interval Size** filter criteria boxes you tick the **Interval Size** tick box, select an operator from the **Operator** drop down box and enter the threshold values in the text boxes.

☞ **Tip:** Instead of ticking the tick box you can click in the text box and as a result the tick box is ticked and the text box is opened for edit.

When you have located the random selection element, select it in the Table Area and then click the **Delete** button on the Toolbar. Alternatively you can select the **Delete** menu item from the **Edit** menu or use the **Delete** key.

9.4 Duplicating Random Selection Configuration Elements

Instead of defining a new random selection configuration element it can sometimes be easier to duplicate an existing element, make the necessary changes and then save it. For further information on how to duplicate refer to section 3.4.2.2 Duplicating Configuration Elements.

9.5 Printing

Printing from the Random Selections tab can be done in two different printouts – you can print the list in the Table Area or you can print random selection details in the Data Area. For information about printing in the Random selection tab refer to section 3.4.2.4 Printing Configuration Elements.

10. Data Loading Tool

ESKORT Clearance Workbench provides a Data Loading Tool which makes the user capable of importing the contents of a structured file into Parameter Tables.

The prerequisites for the system to be able to import the file are that the contents of the file fulfill some formatting requirements:

1. The file must be a plain text file containing text characters as ASCII.
2. The lines in the file must be separated by line feed character (ASCII character 10).
3. An entity which is to be loaded into the Parameter Table must be represented by one line each in the file.
4. The lines to load shall be consecutive, i.e. that there may not be any empty or invalid rows between each row that represents an entity.
5. The last row in the file shall be for loading.
6. Each line in the file represents at least the columns defined as key columns in the Parameter Table in question.
7. In the file each column is separated by a specific symbol.

Figure 115 shows an example of a file fulfilling the requirements.

```
Country Code;Keyword;Value 1;Value 2;My description
AF;"Loaded from Bill's file";1;1;"This is imported from a file to illustrate the function"
AG;"Loaded from Bill's file";2;2;"This is imported from a file to illustrate the function"
AI;"Loaded from Bill's file";3;3;"This is imported from a file to illustrate the function"
AL;"Loaded from Bill's file";3;2;"This is imported from a file to illustrate the function"
AR;"Loaded from Bill's file";2;4;"This is imported from a file to illustrate the function"
```

Figure 115: Example of file to import into a Parameter Table by using the Data Loading Tool.

If the file complies to all the requirements above it can be a source for the loading process.

You can do the data loading by following the steps described in the sections in this chapter.

10.1 Step 1 – Starting the Data Loading Tool

Before you start the Data Loading Tool you must select the Parameter Table in the ESKORT Clearance Workbench into which you would like to the data import, e.g. if you need to import risk levels for the Country Code Risk Object you must navigate to that tab.

Please note that in order to be able to start the Data Loading Tool you must be granted rights to do so by the system administrator.

To start the Data Loading Tool please click on the **Import Data…** menu item located below the **Tools**-menu (see Figure 116). This will start the Data Loading Tool, which is a wizard, i.e. a consecutive series of dialogs which will guide you through the import process.

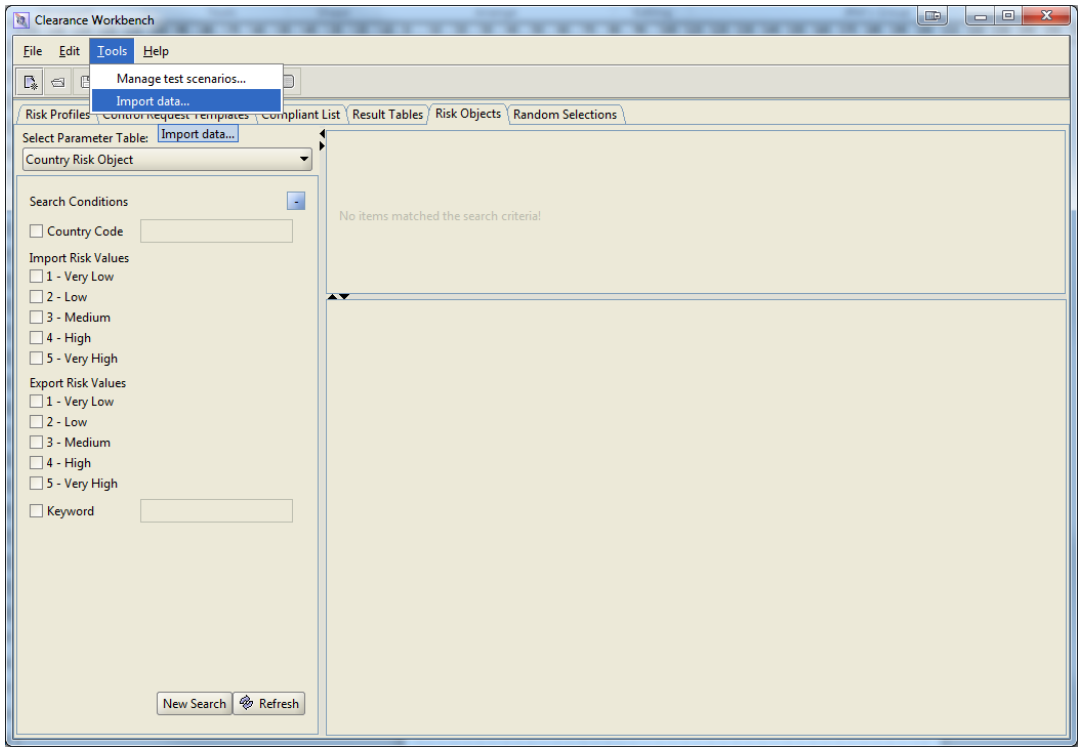


Figure 116: Starting the Data Loading Tool.

10.2 Step 2 – Specifying the Source File

When the Data Loading Tool has been started the next step you must take is to specify the source file from which the data is to be imported. This is shown on the screen shown in Figure 117. On this screen you can type in the path to the file directly into a text box or (the easy way) click the **Browse**-button. Clicking the **Browse**-button opens a familiar file dialog in which you can browse the file system for the source file. When you have selected the file the path to the file is copied into the text box (see Figure 118).

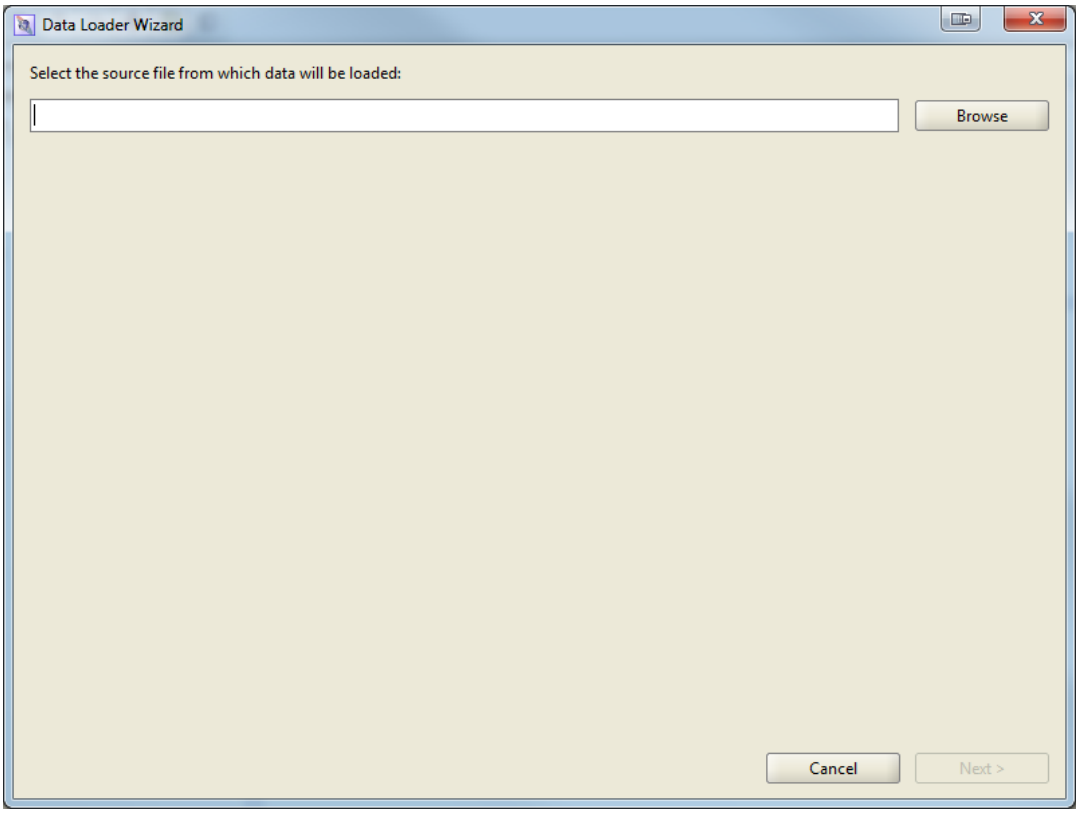


Figure 117: Specification of the source file.

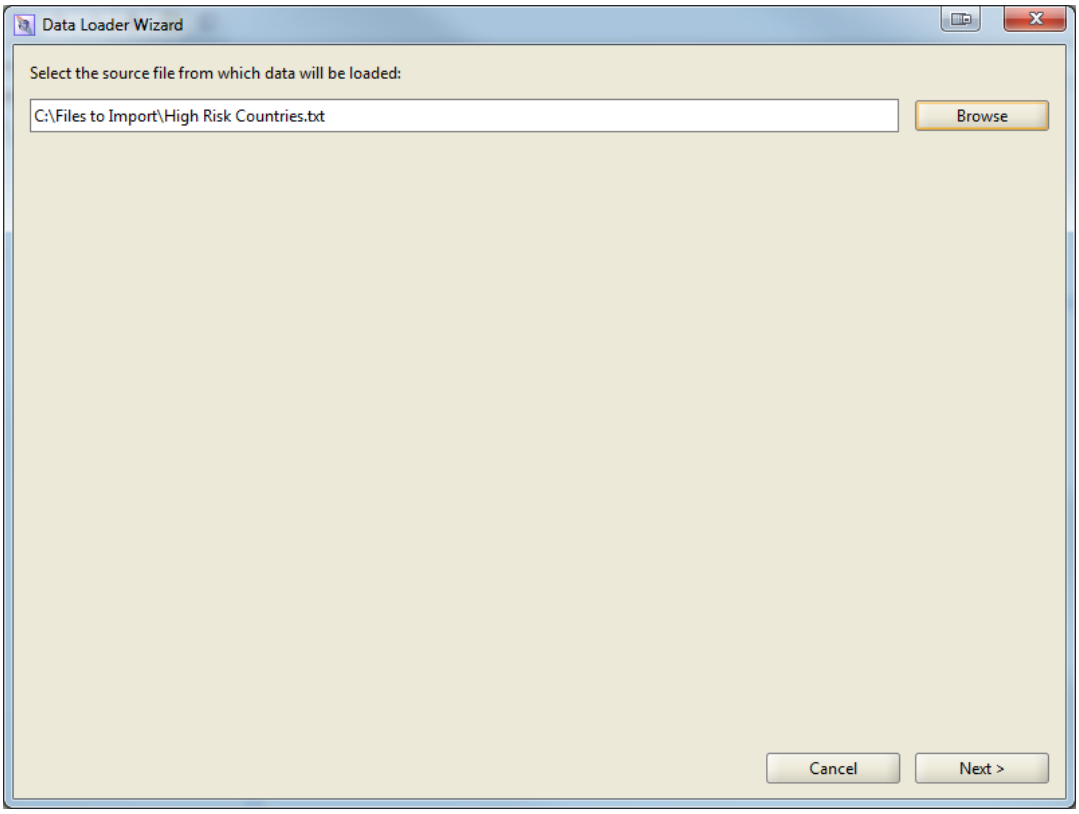


Figure 118: Path to Source File.

Click the **Next >**-button when you have selected to correct source file; hereby a new screen is displayed (see Figure 119).

10.3 Step 3 – Configuring the Interpretation of the Source File

The screen (which is shown in Figure 119) shows how the content of the source file is interpreted by the Data Loading Tool. It does that by displaying the contents in the **Data Preview**-table.

10.3.1 Delimiters

Using the options in the **Delimiters**-group you can specify how the Data Loading Tool shall distinguish between the columns in the source file. You can do that by specifying a delimiter that the Data Loading Tool recognizes as the column separator. A delimiter is a specific symbol such as semicolon, tab, etc. The system offers four standard delimiter symbols, which are the most commonly used for this purpose; however if you need to use another delimiter symbols the Data Loading Tool offers you the capability of entering another (**Other**) symbol.

10.3.2 Consecutive Delimiters

There can be situations where a number of extra delimiters for some reason have been put together in a row. E.g.:

7913;;;"James"

By default the Data Loading Tool will treat each delimiter symbol as the start of a new column; however by checking the **Treat consecutive delimiters as one** check box will suppress this and consecutive delimiters will be interpreted as one.

10.3.3 Text Qualifiers

Some parts in the source file typically contain texts, which are intended to be imported into a text column. If the text for some reason contains one of the selected delimiter symbols the system will interpret the text as two different columns separated by the delimiter symbol. E.g. the text:

My name is Bill; however I am called John

If semicolon is selected to be a delimiter symbol the Data Loading Tool will interpret the text (in the above example) as two columns, where the first column represents the text:

My name is Bill

and the other column represents the text:

however I am called John

Because the intention of the text was that it should be mapped into one single column, but in fact it is mapped into two, it can be the source for errors.

To avoid this, the Data Loading Tool must be configured to tread the text as the source for one single column. This is done by choosing a **Text Qualifier**.

The Data Loading Tool offers you three text qualifiers:

" Quotes

' Apostrophes

Spaces

If the from the example above is encapsulated into quotes, it will look like this:

"My name is Bill; however I am called John"

The two quotes mark the beginning and the end of the text. This enables the Data Loading Tool to ignore the semicolon in fact it is contained inside the two quotes.

10.3.4 Text Qualifiers Used in the Text

In the previous section we learned how to use text qualifiers to make the system distinguish between text and column separators; however reserving a single symbol for that purpose introduces a new issue, i.e. what if we need to use the text qualifier symbol as a part of our text.

Let us assume that we decided to use semicolon as the delimiter symbol and quote as text qualifier. This will handle the text below as intended.

"My name is Bill; however I am called John"

Now let us imagine that we have another text in the file:

"She replied: "I am not sure; however I am positive" to his idea"

This sentence uses the same symbol as we selected as the text qualifier; hence the Data Loading Tool will interpret the sentence as two texts:

She replied: I am not sure

however I am positive to his idea

The four quotes defines two sections of text being encapsulated in quotes (*She replied:* and *to his idea*) and a third (*I am not sure; however I am positive*) in between – not encapsulated by quotes. Because the text in between is not encapsulated and it contains the delimiter the text will interpreted as two columns.

To avoid this text qualifier symbols, which shall not be treated as text qualifiers but as text, shall be stated double. This makes the Data Loading Tool capable of distinguish between when the symbol is used as a text qualifier or as a part of the text.

10.3.5 Start Import at Row

Some text files for import contain introductory text such as name, date and time of creation, columns headings, etc. Lines containing things like that shall not be imported. For that reason you can specify the row number of the first row to import. This is done in the **Start import at row**-control.

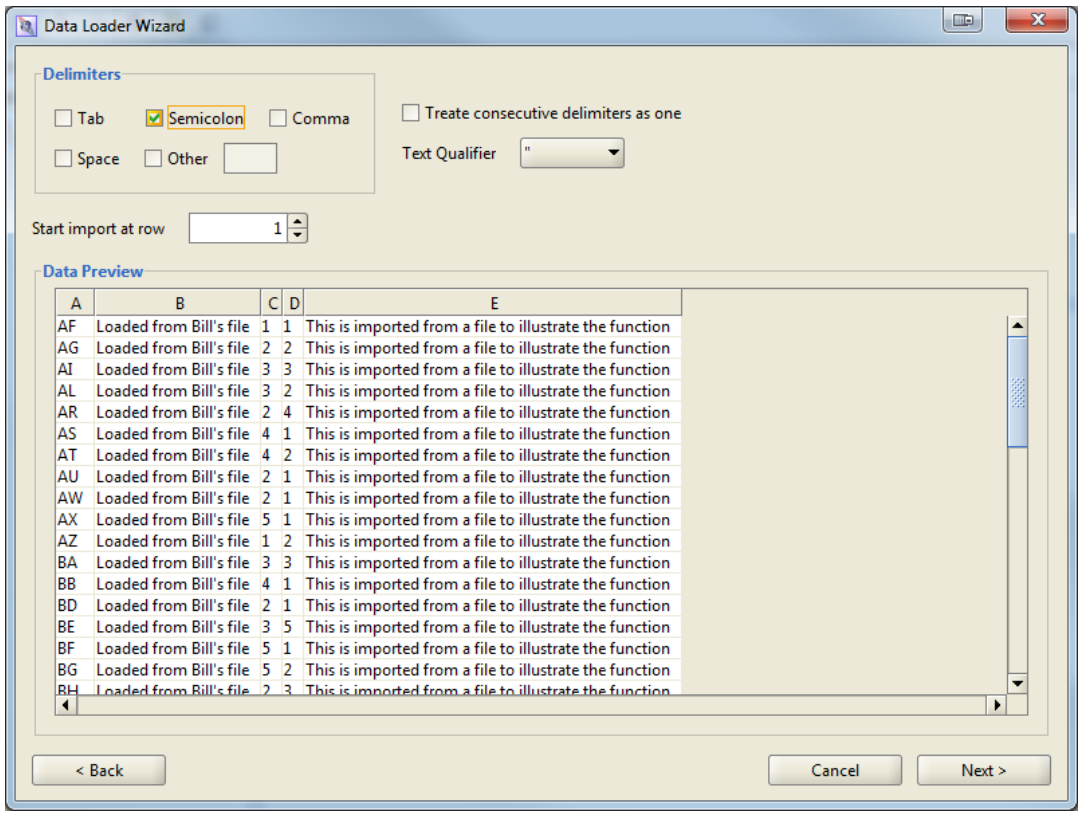

Data Loader Wizard
Delimiters
Tab
Semicolon
Comma
Space
Other
Treate consecutive delimiters as one
Text Qualifier
"
Start import at row
1
Data Preview
A B C D E
AF Loaded from Bill's file 1 1 This is imported from a file to illustrate the function
AG Loaded from Bill's file 2 2 This is imported from a file to illustrate the function
AI Loaded from Bill's file 3 3 This is imported from a file to illustrate the function
AL Loaded from Bill's file 3 2 This is imported from a file to illustrate the function
AR Loaded from Bill's file 2 4 This is imported from a file to illustrate the function
AS Loaded from Bill's file 4 1 This is imported from a file to illustrate the function
AT Loaded from Bill's file 4 2 This is imported from a file to illustrate the function
AU Loaded from Bill's file 2 1 This is imported from a file to illustrate the function
AW Loaded from Bill's file 2 1 This is imported from a file to illustrate the function
AX Loaded from Bill's file 5 1 This is imported from a file to illustrate the function
AZ Loaded from Bill's file 1 2 This is imported from a file to illustrate the function
BA Loaded from Bill's file 3 3 This is imported from a file to illustrate the function
BB Loaded from Bill's file 4 1 This is imported from a file to illustrate the function
BD Loaded from Bill's file 2 1 This is imported from a file to illustrate the function
BE Loaded from Bill's file 3 5 This is imported from a file to illustrate the function
BF Loaded from Bill's file 5 1 This is imported from a file to illustrate the function
BG Loaded from Bill's file 5 2 This is imported from a file to illustrate the function
< Back
Cancel
Next >

Figure 119: The interpretation of the source file.

While you are specifying how the contents of the source file shall be interpreted by the Data Loading Tool you can see the result in the **Data Preview**-table. The contents will change according to the changes you do in the settings. When you are satisfied with the interpretation click the **Next>**-button to continue to the text screen.

10.4 Step 4 – Mapping up the Columns

The purpose of this screen (see Figure 120) is to specify the mapping between the columns specified on the previous screen and the fields in the Parameter Table.

10.4.1 Data Mapping

The **Data Mapping**-table shows fields defined in the selected Parameter Table along with their defined type and the source column, i.e. the columns specified on the previous screen.

In the **Data Mapping**-table you can only change the source columns, the Field Name and Field Type are specified in the definition of the Parameter Table.

By changing the source columns you can decide which columns (specified on the previous screen) that shall be mapped into which fields.

10.4.2 Source Format

All other data types than texts are text being interpreted into a specific type. These types are:

Date

Double

Integer

10.4.2.1 Date Type

The Date Type is a special type which interprets text written in a location specific format into a date.

By default the Data Loading Tool interprets text as a date as if the text is following the following format:

yyyy-MM-dd HH:mm:ss

An example of such a text could be:

1966-07-14 18:15:30

Using the default format the text will be interpreted as: The fourteenth of July the year 1966 at 15 minutes passed 6 in the evening and 30 seconds.

You can however change the way dates are interpreted by setting up the pattern in the **Date and time pattern**-text box. The format of this pattern is described in the next section.

10.4.2.1.1 Date Format Pattern

The following pattern letters are defined:

Letter	Date or Time Component	Presentation	Examples
G	Era designator	Text	AD
y	Year	Year	1996; 96
M	Month in year	Month	July; Jul; 07
w	Week in year	Number	27

Letter	Date or Time Component	Presentation	Examples
W	Week in month	Number	2
D	Day in year	Number	189
d	Day in month	Number	10
F	Day of week in month	Number	2
E	Day in week	Text	Tuesday; Tue
a	Am/pm marker	Text	PM
H	Hour in day (0-23)	Number	0
k	Hour in day (1-24)	Number	24
K	Hour in am/pm (0-11)	Number	0
h	Hour in am/pm (1-12)	Number	12
m	Minute in hour	Number	30
s	Second in minute	Number	55
S	Millisecond	Number	978
z	Time zone	General time zone	Pacific Standard Time; PST; GMT-08:00
Z	Time zone	RFC 822 time zone	-0800

Pattern letters are usually repeated, as their number determines the exact presentation:

- **Text:** Both forms are accepted, independent of the number of pattern letters.
- **Number:** The number of pattern letters is ignored unless it's needed to separate two adjacent fields.
- **Year:** If the number of pattern letters is more than 2, the year is interpreted literally, regardless of the number of digits. So using the pattern "MM/dd/yyyy", "01/11/12" parses to Jan 11, 12 A.D.
  With the abbreviated year pattern ("y" or "yy"), The Data Loading Tool must interpret the abbreviated year relative to some century. It does this by adjusting dates to be within 80 years before and 20 years after the time the interpretation is made using a pattern of "MM/dd/yy" and the Data Loading Tool created on Jan 1, 1997, the text "01/11/12" would be interpreted as Jan 11, 2012 while the text "05/04/64" would be interpreted as May 4, 1964. During parsing, only texts consisting of exactly two digits, will be parsed into the default century. Any other numeric text, such as a one digit text, a three or more digit text, or a two digit text that isn't all digits (for example, "-1"), is interpreted literally. So "01/02/3" or "01/02/003" are parsed, using the same pattern, as Jan 2, 3 AD. Likewise, "01/02/-3" is parsed as Jan 2, 4 BC.
- **Month:** If the number of pattern letters is 3 or more, the month is interpreted as text; otherwise, it is interpreted as a number.
- **General time zone:** Time zones are interpreted as text if they have names. For time zones representing a GMT offset value, the following syntax is used:

```
GMTOffsetTimeZone:
        GMT Sign Hours : Minutes
Sign: one of
        + -
Hours:
```

```
        Digit
        Digit Digit
Minutes:
        Digit Digit
Digit:  one of
            0 1 2 3 4 5 6 7 8 9
```

Hours must be between 0 and 23, and Minutes must be between 00 and 59. The format is locale independent and digits must be taken from the Basic Latin block of the Unicode standard.

For parsing, RFC 822 time zones are also accepted.

- **RFC 822 time zone:** For formatting, the RFC 822 4-digit time zone format is used:

```
RFC822TimeZone:
        Sign TwoDigitHours Minutes
TwoDigitHours:
        Digit Digit
```

TwoDigitHours must be between 00 and 23. Other definitions are as for general time zones.
For parsing, general time zones are also accepted.

The Data Loading Tool also supports *localized date and time pattern* texts. In these texts, the pattern letters described above may be replaced with other, locale dependent, pattern letters.

10.4.2.1.2 Examples

The following examples show how date and time patterns are interpreted in the U.S. locale. The given date and time are 2001-07-04 12:08:56 local time in the U.S. Pacific Time time zone.

Date and Time Pattern	Result
`"yyyy.MM.dd G 'at' HH:mm:ss z"`	`2001.07.04 AD at 12:08:56 PDT`
`"EEE, MMM d, ''yy"`	`Wed, Jul 4, '01`
`"h:mm a"`	`12:08 PM`
`"hh 'o''clock' a, zzzz"`	`12 o'clock PM, Pacific Daylight Time`
`"K:mm a, z"`	`0:08 PM, PDT`
`"yyyyy.MMMMM.dd GGG hh:mm aaa"`	`02001.July.04 AD 12:08 PM`
`"EEE, d MMM yyyy HH:mm:ss Z"`	`Wed, 4 Jul 2001 12:08:56 -0700`
`"yyMMddHHmmssZ"`	`010704120856-0700`
`"yyyy-MM-dd'T'HH:mm:ss.SSSZ"`	`2001-07-04T12:08:56.235-0700`

10.4.2.2 Numeric Types

Double and Integer are numeric types, i.e. they represents numbers with decimals (Double) or without Decimals (Integers).

Dependent on the origin of the source file decimals and thousand separators are indicated differently, e.g. in Denmark dot (.) is used as thousand separator and comma (,) is used as decimal separator. The opposite is the case in the United States.

The Data Loading Tool supports both, but only one at a time. You specify the way the Data Loading Tool shall interpret the decimals by selecting either a using the **Decimal separator**-combo box.

10.4.3 Data Preview

While you are changing the source columns you will immediately see the effect in the **Data Preview**-table.

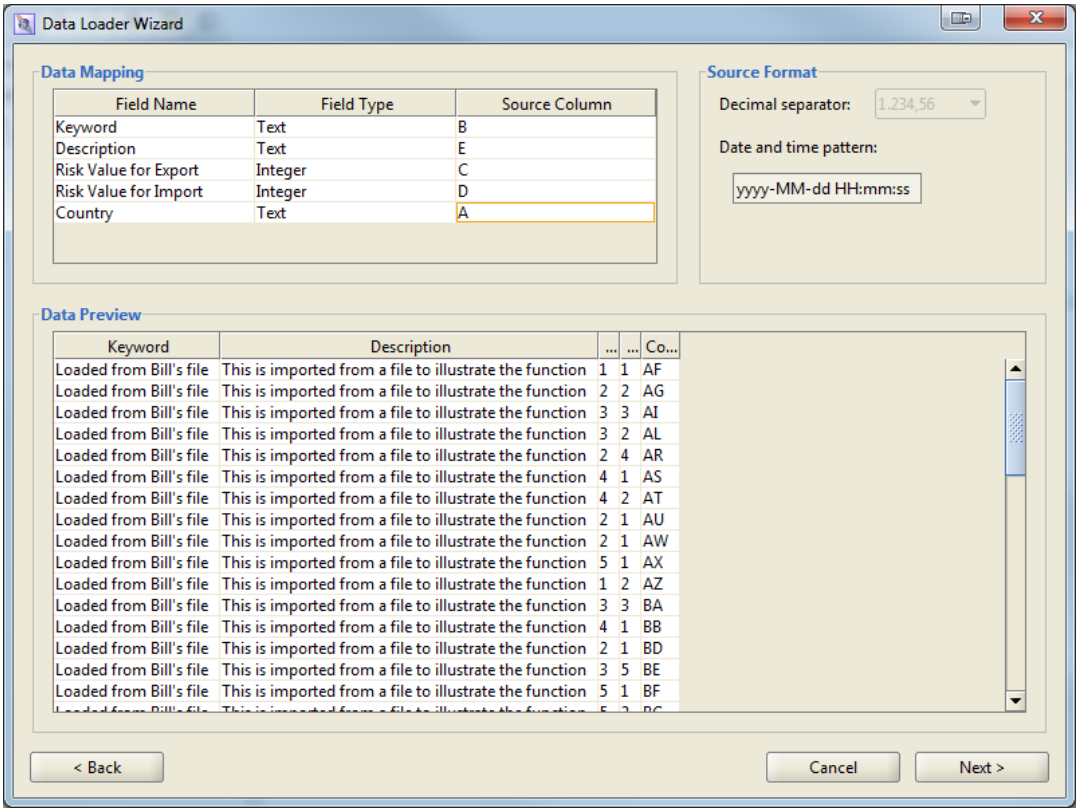


Figure 120: Mapping between the columns and the fields.

Click the **Next>**-button when you find that the Data Preview shows the correct interpretation of your source file. This will bring you to the next screen.

10.5 Step 5 – Starting the Import

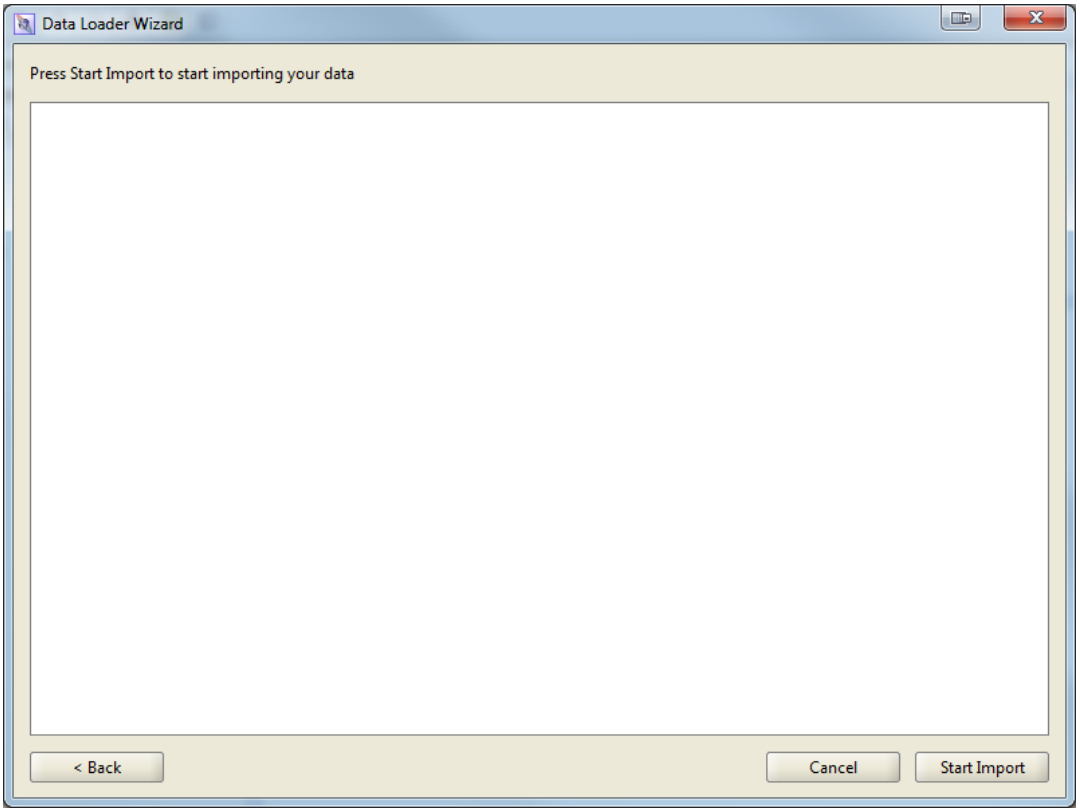


Figure 121: Screen Starting the Import

This screen is the one that is displayed just before, under and after the data loading process (import) has been started. The screen mainly contains a huge text area in which information from the import is logged.

To initiate the import please click the **Start Import**-button. This will start the import and the contents of the screen will change to the screen displayed in Figure 122.

The screen at Figure 122 shows the progress of the import and possibly information regarding errors etc.

While the import is running you can cancel it by clicking the **Stop Import**-button. Otherwise the import will continue until all rows in the source file has been loaded and the process will end.

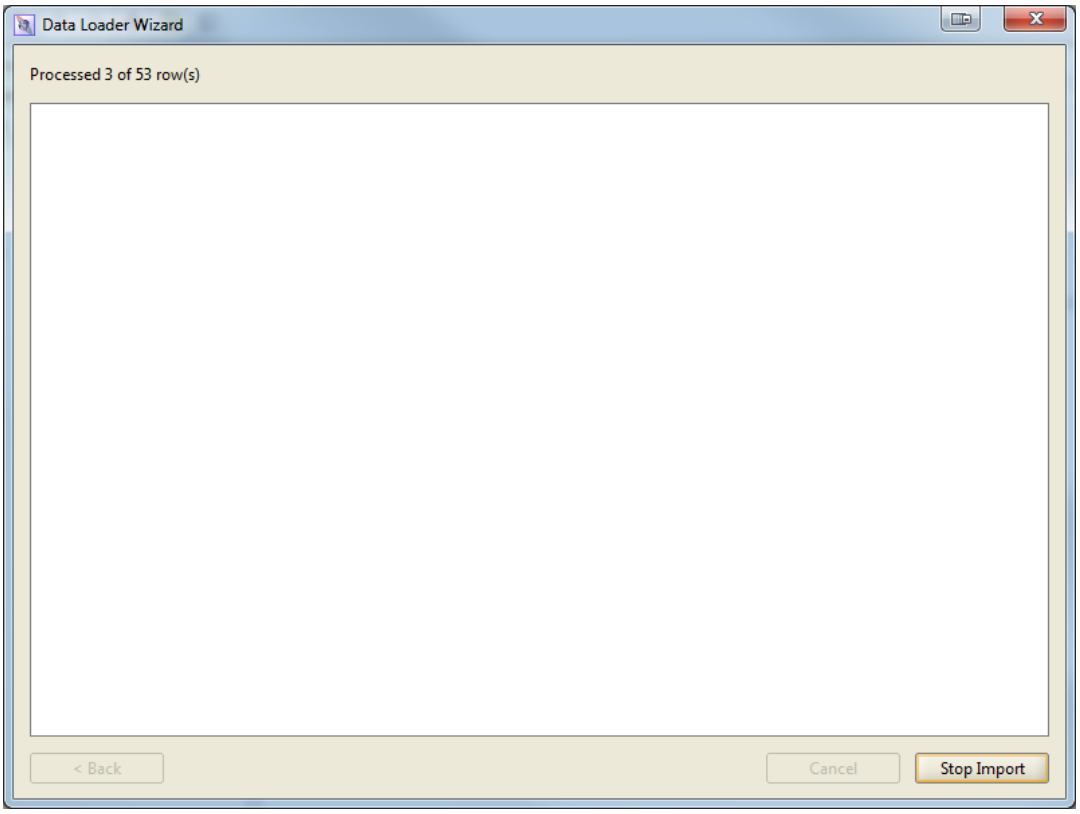


Figure 122: Screen when import process is running.

10.6 Step 6 – Import Reporting

When the import process has completed a report is written into the log text pane as shown in Figure 123.

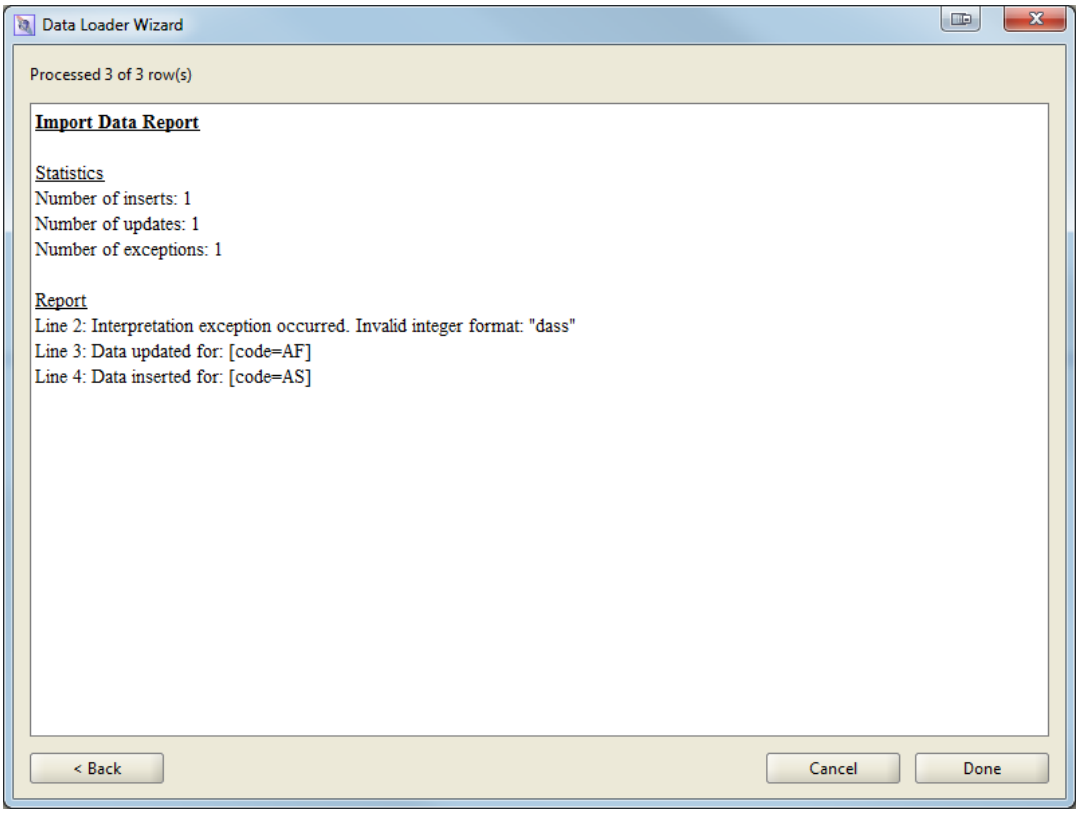


Figure 123: Data Loading Tool after completed import process.

Clicking the **Done**-button will end the Data Loading Tool and you will be back again in the ESKORT Clearance Designer.

10.7 Step 7 – After the Import

When the Data Loading Tool has completed you will be able to find the imported data in the list view of the Parameter Table in question. An example of this is shown in Figure 124.

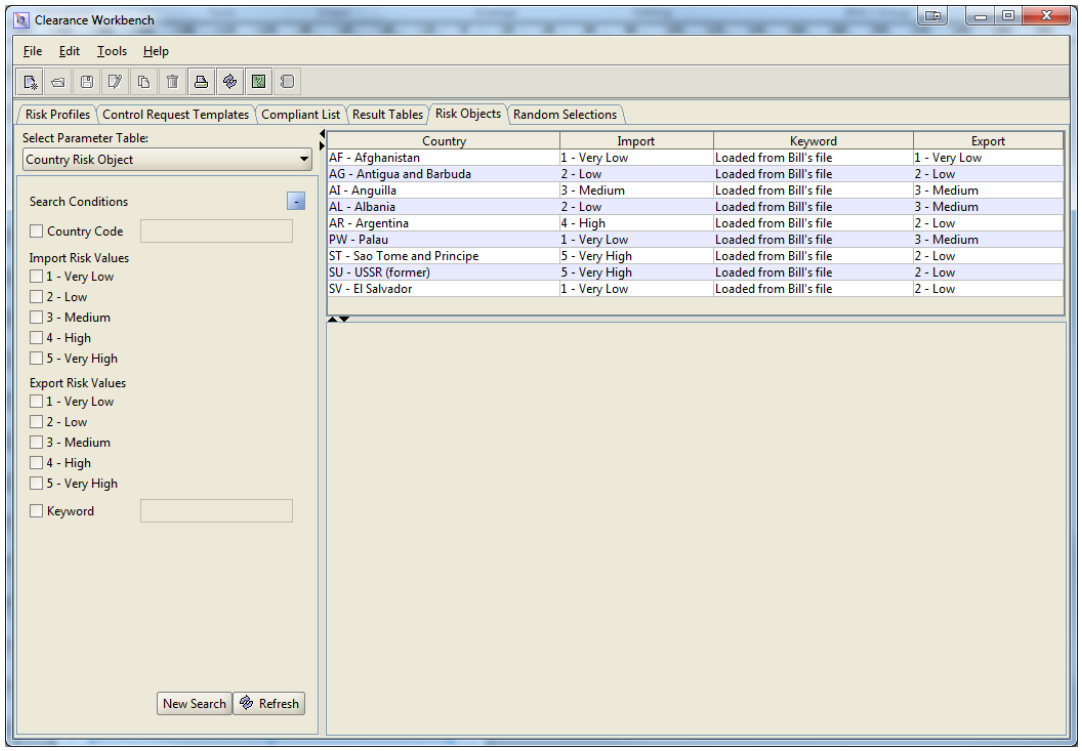


Figure 124: Contents of a Parameter Table after import.

11. Appendix 1: MDC Language Definition

The following provides a complete overview of the MDC Language Grammar. The grammar is presented in EBNF, using the following conventions:

Style	Meaning
<symbol>	Indicates a non-terminal symbol of the grammar (e.g. <area> or <rule>).
keyword	Indicates a keyword, syntactical symbol or operator (e.g. **within** or **and**).
string	Indicates a terminal symbol (e.g. identifier, string, real or unsigned-integer).
'empty'	Indicates the empty production.

Contents:

- Grammar
- Terminal Symbols

11.1 Grammar

list ::=

    'empty'
    <areas>

areas ::=

    <area>
    <areas> <area>

area ::=

    area <new_top_area> { <areabody> }

derived_area ::=

    area <new_derived_area> { <areabody> }

new_top_area ::=

    'empty'
    identifier

new_derived_area ::=

    'empty'
    identifier

areabody ::=

    'empty'
    <area_head_members> <areamembers>

area_head_members ::=

    'empty'
    <area_head_members_ne>

area_head_members_ne ::=

<area_head_member>
<area_head_members_ne> <area_head_member>

area_head_member ::=

within <withinbody>
let <letbody>
filter <filterbody>

areamembers ::=

<areamember>
<areamembers> <areamember>

areamember ::=

rule <rulescope>
<derived_area>

withinbody ::=

(<cubenames> **) for** <forbody>

forbody ::=

(<sets> **)**

identifier ::=

<regular_identifier>

regular_identifier ::=

identifier

cubenames ::=

<cubename>
<cubenames> **,** <cubename>

cubename ::=

<identifier>

dimension ::=

<identifier>

qualified_dimension ::=

<cubename> **:** <dimension>

member ::=

<identifier>
<dimension> **.** <identifier>
<dimension> **. FIRSTMEMBER**
<dimension> **. LASTMEMBER**
<dimension> **. PREVMEMBER**
<dimension> **. NEXTMEMBER**
<qualified_dimension> **.** <identifier>
<cubename> **:** <identifier>

<member_value_expression>

new_member ::=

<member>

members ::=

<member>
<members> , <member>

tuple ::=

{ <members> }
<member>

tuples ::=

<tuple>
<tuples> , <tuple>

set ::=

<set_value_expression>
{ <tuples> }

sets ::=

<set>
<sets> , <set>

letbody ::=

<new_member> = <value_expression>

filterbody ::=

<filter_id> { <filter_set> }

filter_id ::=

identifier

filter_set ::=

<filter_element>
<filter_set> , <filter_element>

filter_element ::=

unsigned-integer
string
<filter_range_start> - <filter_range_end>

filter_range_start ::=

unsigned-integer

filter_range_end ::=

unsigned-integer

value_expression ::=

<numeric_value_expression>

numeric_value_expression ::=

<value>

value ::=

<term>
<value> + <term>
<value> - <term>

term ::=

<factor>
<term> * <factor>
<term> / <factor>

factor ::=

<numeric_primary>
+ <numeric_primary>
- <numeric_primary>

numeric_primary ::=

<value_expression_primary>
<numeric_value_function>

value_expression_primary ::=

<unsigned_numeric_literal>
<character_string_literal>
(<value_expression>)
<tuple>
MEMBER_NAME (<member>)

unsigned_numeric_literal ::=

<exact_numeric_literal>

character_string_literal ::=

string

exact_numeric_literal ::=

real
unsigned-integer

rulescope ::=

<rulebody> <end_rule>
<start_named_rule> <rulebody> <end_rule>

start_named_rule ::=

<rule_identifier>

end_rule ::=

.

rule_identifier ::=

identifier

rulebody ::=

if <search_condition> **then** <action_list>

search_condition ::=

<boolean_term>
<search_condition> || <boolean_term>

boolean_term ::=

<boolean_factor>
<boolean_term> **&&** <boolean_factor>

boolean_factor ::=

<boolean_primary>
not <boolean_primary>

boolean_primary ::=

<value_expression> <comp_op> <value_expression>
(<search_condition> **)**
<function_identifier> **(** <param_list> **)** <function_end>

numeric_value_function ::=

SUM (<set> **)**
AVG (<set> **)**
<function_identifier> **(** <param_list> **)** <function_end>

function_identifier ::=

identifier

comp_op ::=

=
<>
>
<
>=
<=

set_value_expression ::=

<dimension>
<dimension> **. MEMBERS**
<dimension> **. EXCEPT (** <member> **)**

member_value_expression ::=

<dimension> **.**
<dimension> **. CURRENTMEMBER**

action_list ::=

'empty'
<action>
<action_list> **;** <action>

action ::=

call <oleidentifier> **(** <param_list> **)**

oleidentifier ::=

identifier

param_list ::=

'empty'
<param_list_ne>

param_list_ne ::=

<parameter>
<param_list_ne> **,** <parameter>

parameter ::=

<value>

11.2 Terminal Symbols

The grammar is based on the following terminal symbols:

identifier

Any sequence of aplha-numeric characters starting with a non-numeric character;
(e.g. MyName, First_Name).

string

Any sequence of aplha-numeric characters delimited with "" characters;
(e.g. "This is a string", or "This is \"a string\"").

unsigned-integer

An unsigned integer number constant;
(e.g. MyName, First_Name).

real

A decimal number constant;
(e.g. 1000000 or 0.5).

12. Appendix 2: Function Reference

The following provides a complete overview of the functions currently available in the MDC language.

12.1 Special argument types

Some of the functions take argument types that may seem to be unfamiliar to most users. This section shortly describes these kinds of argument types.

12.1.1 TUPLE (a.k.a. COORDINATE)

The TUPLE type designates a value described as a "tuple" in the sense of the MDC grammar, that is, a value that adhere to the definition of a tuple in the MDC grammar . Examples could be:

{MyCube:my_field} or *{my_field}* or simply *my_field*

since all of these examples matches the grammar of a tuple. Thus, a function that takes a TUPLE as argument type accepts a fieldname as argument but NOT a simple string value or simple number value.

Note: The TUPLE as type is also sometimes referred to as COORDINATE.

12.1.2 ANY_TYPE (a.k.a UNKNOWN)

The function argument type ANY_TYPE indicates that the function accepts any of the simple types like boolean, string, double or long.

Note: The ANY_TYPE as type is also sometimes referred to as UNKNOWN.

12.2 Functions

12.2.1 AbsValue (double x)

Description: Returns the absolute value of the parameter (*x*). For positive values and zero the result will be the same as the parameter (*x*). For negative values the result will be the positive value (the minus sign will be removed), i.e. *–x*.

Syntax: double AbsValue(double x)

Parameters:

x The double value which absolute value is returned.

Example:

```
AbsValue(-1000)
```

Returns 1000

```
AbsValue(3.14)
```

Returns 3.14

12.2.2 ActivateArea (string AreaName)

Description: Activates the area (or rule) specified by the parameter.

Syntax: void ActivateArea(string Area*Name*)

Parameters:

AreaName The name of the area (or rule) which are activated.

Example:

```
ActivateArea("MyArea")
```

Activates the area (or rule) named “MyArea”.

12.2.3 AllEmpty (TUPLE Measure, string DimensionName)

Description: Returns ***TRUE*** if all cells identified by the measure (*Measure*) are empty for the dimension (*DimensionName*).

Syntax: boolean AllEmpty(TUPLE *Measure*, string *DimensionName*)

Parameters:

Measure	The measure member which are investigated for a specified dimension (*DimensionName*)**.**
DimensionName	The name of the dimension in which the member's existence is investigated.

Example:

```
AllEmpty(TaxableIncome, "Years")
```

Returns ***TRUE*** if no measures in the years-dimension exist. If one or more exists the function returns ***FALSE***.

12.2.4 AllZeroOrNotKnown (TUPLE Measure)

Description: Returns ***TRUE*** when all cells built with the measure (*Measure*) and all the ordinary members of the period dimension are NULL/EMPTY or zero.

Syntax: boolean AllZeroOrNotKnown(TUPPLE Measure)

Parameters:

Measure	The measure that the function determines whether all the measures are not known or is zero.

Example:

```
AllZeroOrNotKnown(TaxableIncome)
```

Returned ***TRUE*** if no declared taxable incomes are not known by the system or if all the declared incomes are zero; otherwise ***FALSE*** is returned.

See also:
IsEmpty
IsKnown
IsKnownAndNotZero
IsNotKnown

12.2.5 AnyZeroOrNotKnown (TUPLE Measure)

Description: Returns ***TRUE*** when any cells built with the measure (*Measure*) and all the ordinary members of the period dimension are NULL/EMPTY or zero.

Syntax: boolean AnyZeroOrNotKnown(TUPPLE Measure)

Parameters:

Measure The measure that the function determines whether any of the measures are not known or are zero.

Example:

```
AnyZeroOrNotKnown(TaxableIncome)
```

Returned ***TRUE*** if any declared taxable incomes are known as zero by the system or if any the declared incomes are unknown; otherwise ***FALSE*** is returned.

See also:
IsEmpty
IsKnown
IsKnownAndNotZero
IsNotKnown

12.2.6 Assign (TUPLE Tuple, ANY_TYPE Value)

Description: Sets the value (*Value*) in the cell identified by the tuple (*Tuple*).

Syntax: void Assign(TUPLE *Tuple*, ANY_TYPE *Value*)

Parameters:

Tuple The coordinate of the cell into which the value is assigned.

Value The value that is assigned into the cube.

Example:

```
Assign(StatusCode, "1")
```

Sets the *StatusCode* member in the current context to the value "1".

12.2.7 Average (TUPLE Measure, string DimensionName)

Description: Returns the average value of measure *Measure* for the dimension *DimensionName.*

Syntax: double Average(TUPLE *Measure*, string *DimensionName*)

Parameters:

Measure	The measure member which average will be calculated over the specified dimension (*DimensionName*)**.**
DimensionName	The name of the dimension over which the average is calculated.

Example:

```
Average(TaxableIncome, "Years")
```

Returns the average taxable income over the years.

12.2.8 BuildDate (ANY_TYPE date, string modification)

Description: Builds a date by applying one or more modification steps (separated by semicolons) to the base date (*date*) passed. The passed base date and the returned date will be in the format *yyyymmdd* or *yyyy-mm-dd* (determined by the length of the string).

The following step types are supported:

- [*y*|*q*|*m*|*d*]<*count*> will add *count* years (*y*), quarters (*q*), months (*m*) or days (*d*) respectively.
- *fd*[[*y*|*q*|*m*] will change date to the first day of year (*fdy*), quarter (*fdq*) and month (*fdm*) respectively.

ld[[*y*|*q*|*m*] will change date to the last day (*ld*) of year, quarter (*ld*) and month (*lm*) respectively.

Syntax: string BuildDate(ANY_TYPE date, string modification)

Parameters:

date	The base date that is modified.
modification	Patterns describing the modification applied.

Example:

```
BuildDate(LatestVATReturnDate, "m:-5;fdm")
```

12.2.9 ConcatStr (string s_1, string s_2)

Description: Returns a string which contents is the concatenation of the two argument strings (s_1 and s_2)

Syntax: string ConcatStr(string s1, string s2)

Parameters:

s1 The string which will form the left part of the concatenated string

s2 The string which will form the right part of the concatenated string

Example:

```
ConcateStr("The first string", "The second string")
```

Returns the text "The first stringThe second string"

12.2.10 CountTo (integer n)

Description: Increases a general counter. If the counter reaches the specified threshold (*n*) then the counter is reset and the function returns ***TRUE***. If the threshold is not reached then the function returns ***FALSE***.

Hereby it is possible to make a rule which "fires" with a fixed specified interval.

Syntax: boolean CountTo(integer n)

Parameters:

n The threshold at which the counter reset and the function returns ***TRUE***.

Example:

```
CountTo(36)
```

Returns the ***TRUE*** one out of 36 calls. The other 35 calls returns ***FALSE***.

12.2.11 CountToWithKey (integer n, string key)

Description: Increases the counter identified by the specified key (*key*). If the counter reaches the specified threshold (*n*) then the counter is reset and the function returns ***TRUE***. If the threshold is not reached then the function returns ***FALSE***.

Hereby it is possible to make a rule which "fires" with a fixed specified interval.

By giving the counter a key it is possible to have multiple counters configured each having a current count and threshold.

Syntax: boolean CountToWithKey(integer n, string key)

Parameters:

n The threshold at which the counter reset and the function returns ***TRUE***.

key The name by which the counter is identified. If the key is an empty string (or NULL) the function will return **FALSE**.

Example:

```
CountToWithKey(36, "MyCounter")
```

Returns the ***TRUE*** one out of 36 calls by the key "MyCounter". The other 35 calls returns ***FALSE***.

12.2.12 CubeEmpty (string CubeName)

Description: Returns ***TRUE*** if the entire cube (*CubeName*) is empty.

Syntax: boolean CubeEmpty(string *CubeName*)

Parameters:

CubeName The name of the cube which is investigated.

Example:

```
CubeEmpty("MyCube")
```

Returns ***TRUE*** if the cube named MyCube is empty; otherwise ***FALSE*** is returned.

12.2.13 DateHalfAYearBack (string date)

Description: Returns the date that is half a year earlier than the specified date (*date*).

Both the specified date and the returned date has the format: *YYYYMMDD*

The specified date is a string which is assumed to hold a date in the leftmost 8 positions. This means that valid arguments are of a form *YYYYMMDD* or *YYYYMMDD-YYYYMMDD*.

If the specified date is an empty string, then the current date is used.

Syntax: string DateHalfAYearBack("19660714")

Parameters:

date The date which is used as offset for determining the date a half year back.

Example:

```
DateAHalfYearBack(MEMBER_NAME(CalendarYears.CURRENTMEMBER))
```

12.2.14 DateToLong (string date)

Description: Converts the specified date (*date*) from a string to an integer value (*n*) calculated as:

$$n = 10000 \cdot YYYY + 100 \cdot MM + DD$$

If *date* specifies an invalid date then 0 is returned.

Syntax: long DateToLong("19660714")

Parameters:

date The date which is converted into an integer value. The *date* must be a string of either the length 8 or 10.

If the string is 8 characters long the date must be specified at the following format:

YYYYMMDD

If the string is 10 character long the date must be specified at the following format:

YYYY□MM□DD, where □ can be any character.

Example:

```
DateToLong("2010x05x04")
```

Returns the integer value 20100504

12.2.15 DateXYearBack (string date, double nYears)

Description: Returns the date that is the specified number of years (*nYears*) earlier than the specified date (*date*).

The returned date has the format: *YYYYMMD*D.

If *date* is an empty string, then current date is used instead.

Syntax: long DateXYearBack(string date, double nYears)

Parameters:

date The date from which the number of years are withdrawn before the final date is returned.

The *date* argument is a string which is assumed to hold a date in the leftmost 8 positions. This means that valid arguments are of a form *YYYYMMDD* or *YYYYMMDD-YYYYMMDD*.

Example:

```
DateXYearBack("20100504", 3)
```

Returns the integer value 20100504

12.2.16 DaysBetween (string startDate, string endDate)

Description: Returns the number of days between the two specified dates. If the two dates are equal 0 is returned.

Syntax: double DaysBetween(string startDate, string endDate)

Parameters:

startDate The date from which the number of days are counted. The argument is a string which is assumed to hold a date in the leftmost 8 positions. This means that valid arguments are of a form *YYYYMMDD* or *YYYYMMDD-YYYYMMDD*.

endDate The date until which the number of days are counted. The argument is a string which is assumed to hold a date in the leftmost 8 positions. This means that valid arguments are of a form *YYYYMMDD* or *YYYYMMDD-YYYYMMDD*.

Example:

```
DaysBetween("20100504", "20100501")
```

returns the value 3.

12.2.17 DaysPassed (string since)

Description: Returns the number of days that has passed since the specified date (*since*). If the specified date is the current day then 0 is returned.

Syntax: double DaysPassed(string since)

Parameters:

since The date from which until now the number of days are counted. The argument is a string which is assumed to hold a date in the leftmost 8 positions. This means that valid arguments are of a form *YYYYMMDD* or *YYYYMMDD-YYYYMMDD*.

Example:

```
DaysPassed("19660714")
```

Returns the number of days since July 14 1966.

12.2.18 DeActivateArea (string AreaName)

Description: Deactivates the area (or rule) specified by the parameter.

Syntax: void DeActivateArea(string Area*Name*)

Parameters:

AreaName The name of the area (or rule) which are deactivated.

Example:

```
DeActivateArea("MyArea")
```

Deactivates the area (or rule) named "MyArea".

12.2.19 DblToLng (double x)

Description: Converts a decimal value (*x*) into an integer value by truncating (flooring) the decimal value.

Syntax: long DblToLng(double x)

Parameters:

x The decimal value that is converted into an integer value.

Example:

```
DblToLng(turnover/nEmployee)
```

Returns the turnover per employee truncated into an integer value.

12.2.20 DblToStr (double x)

Description: Converts a decimal value (*x*) into a string (a text).

Syntax: string DblToStr(double x)

Parameters:

x The decimal value that is converted into a string.

Example:

```
DblToStr(3.14)
```

Returns the number 3.14 as the text "3.14"

12.2.21 Difference (double x1, double x2)

Description: Returns the difference between x_1 and x_2, i.e. the value calculated as $x_2 - x_1$.

Syntax: double Difference(double x_1, double x_2)

Parameters:

x_1 The value to which the difference to x_2 is calculated.

x_2 The value to which the difference from x_1 is calculated.

Example:

```
Difference(5.5, 7)
```

Returns the number 7 – 5.5. = 1.5

```
Difference(13, 11.3)
```

Returns the number 11.3 – 13 = – 1.7

12.2.22 EndDate (string period)

Description: Returns the end date of the specified period (*period*) as an integer value.

Syntax: long EndDate(string period)

Parameters:

period A period that complies to the specified pattern:
YYYYMMDD-YYYYMMDD.

Example:

```
EndDate("19660714-20100504")
```

Returns the integer value 20100504

12.2.23 ExecuteArea (string AreaName)

Description: Executes the area (or rule) specified by the parameter. When the specified area (or rule) has been executed the evaluation will continue from the calling point. The area can be located anywhere inside the knowledge base.

Syntax: void ExecuteArea(string Area*Name*)

Parameters:

AreaName The name of the area (or rule) which are executed.

Example:

```
ExecuteArea("MyArea")
```

Executes the area (or rule) named "MyArea".

12.2.24 FilterString (string s, string trimChars)

Description: Returns the contents of the specified string (*s*) but where all the occurrences of the characters in the specified string (*trimChars*) have been removed.

Syntax: string FilterString(string s, string trimChars)

Parameters:

s The string which are filtered.

trimChars A string specifying the characters which shall be removed from *s*.

Example:

```
FilterString("Intracom IT-Services", " IT")
```

Returns the text "ntracom-Services"

12.2.25 FormatNumber (double value, string format)

Description: Formats the specified value (*value*) into a string using the specified format.

Syntax: string FormatNumber(double value, string format)

Parameters:

value The value that is formatted into a string

format A string specifying the format according to Appendix F.

Example:

```
FormatNumber(7/3, "Amount=%N")
```

Returns the text "Amount=2"

12.2.26 GetCubeName (CURRENT_CUBE)

Description: Returns a string containing the name of the specified cube (*cube*). Hereby it is possible to get the name of a cube into a string which then can be used rules.

Syntax: string GetCubeName(OBJECT cube)

Parameters:

CURRENT_CUBE The keyword CURRENT_CUBE. Note that this is a keyword of the language and must be spelled exactly like that. This is the only valid input to this argument and specifies the current cube which name should be returned.

Example:

```
GetCubeName(CURRENT_CUBE)
```

Returns a string containing the text: "IncomeTax" if current cube is IncomeTax.

12.2.27 GetNoOfMembers (CURRENT_CUBE, string dimensionName)

Description: Returns the number of members in the dimension specified by the second parameter *dimensionName*.

Syntax: double getNoOfMembers(CURRENT_CUBE, string DimensionName)

Parameters:

CURRENT_CUBE The keyword CURRENT_CUBE. Note that this is a keyword of the language and must be spelled exactly like that. This is the only valid input to this argument!

This argument has no effect on the result but is retained for backward compatibility and must be supplied.

dimensionName The name of the dimension which members are counted.

Example:

```
GetNoOfMembers(CURRENT_CUBE, "Years")
```

Returns the number of members in the *Years*-dimension in the current cube.

12.2.28 GetParamDoubleValue(string UsageName)

Description: Returns the result from the specified Parameter Table Usage as a double value.

Syntax: double GetParamDoubleValue(string Usage*Name*)

Parameters:

UsageName The name of the usage which are called.

Example:

```
GetParamDoubleValue("MyDoubleUsage")
```

Calls the usage named "MyDoubleUsage" and returns the double value returned by the usage.

12.2.29 GetParamIntValue (string UsageName)

Description: Returns the result from the specified Parameter Table Usage as an integer value.

Syntax: integer GetParamIntValue(string Usage*Name*)

Parameters:

UsageName The name of the usage which are called.

Example:

```
GetParamIntValue("NumberOfTickets")
```

Calls the usage named "`NumberOfTickets`" and returns the integer value returned by the usage.

12.2.30 GetParamStrValue (string UsageName)

Description: Returns the result from the specified Parameter Table Usage as a string.

Syntax: string GetParamStrValue(string Usage*Name*)

Parameters:

UsageName The name of the usage which are called.

Example:

```
GetParamStrValue("StandardTaxOfficeName")
```

Calls the usage named "`StandardTaxOfficeName`" and returns the string returned by the usage.

12.2.31 GetRandomNo (double min, double max)

Description: Returns a pseudo random number within the range limited by the specified values (*min* as the lower limit and *max* the upper limit).

Syntax: double GetRandomNo(double min, double max)

Parameters:

min The lower limit of the range within in which the pseudo random number will be given.

max The upper limit of the range within in which the pseudo random number will be given.

Example:

```
GetRandomNo(20, 30)
```

Returns a number between 20 and 30.

12.2.32 InString (string candidate, string s)

Description: Returns ***TRUE*** if the specified string (*candidate*) is contained in the specified string (*s*); Otherwise ***FALSE*** is returned.

Syntax: boolean InString(string candidate, string s)

Parameters:

candidate The string for which *s* is searched

s The string which is searched for the occurrence of *candidate*.

Example:

```
InString("Abe", "Abe Lincoln")
```

Returns true because "Abe" is contained in "Abe Lincoln"

12.2.33 IsEmpty (TUPLE Measure)

Description: Determines if the specified measure (*Measure*) is empty. If it is empty then ***TRUE*** is returned; otherwise ***FALSE***.

Syntax: boolean IsEmpty(TUPPLE Measure)

Parameters:

Measure The measure that the function determines whether it is empty.

Example:

```
IsEmpty(TaxableIncome)
```

Returned ***TRUE*** if no taxable income is declared; otherwise ***FALSE*** is returned.

See also:

AllZeroOrNotKnown
IsKnow
IsKnownAndNotZero
IsNotKnown
IsZeroOrNotKnown

12.2.34 IsKnown (TUPLE Measure)

Description: Determines if the specified measure (*Measure*) is known (a.k.a. NULL/EMPTY – I.e. has been assigned a value during data extraction or assignment). If it is known then ***TRUE*** is returned; otherwise ***FALSE***.

Syntax: boolean IsKnown(TUPPLE Measure)

Parameters:

Measure The measure that the function determines whether it is known.

Example:

```
IsKnown(TaxableIncome)
```

Returned ***TRUE*** if no taxable income is known by the system; otherwise ***FALSE*** is returned.

See also:

AllZeroOrNotKnown
IsEmpty
IsKnownAndNotZero
IsNotKnown
IsZeroOrNotKnown

12.2.35 IsKnownAndNotZero (TUPLE Measure)

Description: Determines if the specified measure (*Measure*) is known (a.k.a. not NULL/EMPTY – I.e. has been assigned a value during data extraction or assignment). If it is not known then ***FALSE*** is returned; otherwise the function returns ***TRUE*** if the value is not zero; otherwise it returned ***FALSE***.

Syntax: boolean IsNotKnown(TUPPLE Measure)

Parameters:

Measure The measure that the function determines whether it is known and not zero.

Example:

```
IsKnownAndNotZero(TaxableIncome)
```

Returned ***FALSE*** if no taxable income is known by the system; otherwise ***TRUE*** is returned if the taxable income is different from zero.

See also:

AllZeroOrNotKnown
IsEmpty
IsKnown
IsNotKnown
IsZeroOrNotKnown

12.2.36 IsNotKnown (TUPLE Measure)

Description: Determines if the specified measure (*Measure*) is not known (a.k.a. NULL/EMPTY – I.e. has not been assigned a value during data extraction or assignment). If it is not known then ***TRUE*** is returned; otherwise ***FALSE***.

Syntax: boolean IsNotKnown(TUPPLE Measure)

Parameters:

Measure The measure that the function determines whether it is not known.

Example:

```
IsKnown(TaxableIncome)
```

Returned ***FALSE*** if no taxable income is known by the system; otherwise ***TRUE*** is returned.

See also:

AllZeroOrNotKnown
IsEmpty
IsKnown
IsKnownAndNotZero
IsZeroOrNotKnown

12.2.37 IsStandardContainerCode (string code)

Description: Interprets the format of the specified container code (*code*). If the container code represents what is considered to be a standard container code then the function returns ***TRUE***; otherwise ***FALSE*** is returned.

As shown in the table below, all prefix numbers have their own values. The value of a number in the container code is equal to the number itself.

Prefix number	Value
A	10
B	12
C	13
D	14
E	15
F	16
G	17
H	18
I	19

Prefix number	Value
J	20
K	21
L	23
M	24
N	25
O	26
P	27
Q	28
R	29

Prefix number	Value
S	30
T	31
U	32
V	34
W	35
X	36
Y	37
Z	38

Values of Prefix Number

The last, seventh number (placed on the container in a small square) is not included, but is the last digit of the calculation and by this the control digit. For example, take the container number:

MWCU 605978-4.

The pre-fix MWCU has the following value:

M = 24
W = 35
C = 13
U = 32

With this the following calculation is made, by which every value is multiplied with twice the number of which the previous value was multiplied (starting with the digit 1).

M	24	x	1	=	24
W	35	x	2	=	70
C	13	x	4	=	52
U	32	x	8	=	256
6	6	x	16	=	96
0	0	x	32	=	0
5	5	x	64	=	320
9	9	x	128	=	1152

7	7	x	256	=	1792
8	8	x	512	=	4096
					7858

The result of the calculation is then divided by eleven and the remainder of the division is the last digit.

```
7858 : 11 = 714
77
 15
 11
 48
 44
  4
```

With shipper owned containers completely different digit combinations are being used, such that this calculation will often give a false outcome.

Syntax: boolean IsStandardContainerCode(string code)

Parameters:

code The code that is interpreted.

Example:

```
IsStandardContainerCode("MWCU 605978-4")
```

Returns the ***TRUE*** because the specified container code is a standard container code.

12.2.38 IsZeroOrNotKnown (TUPLE Measure)

Description: Determines if the specified measure (*Measure*) is not known (a.k.a. NULL/EMPTY – I.e. has not been assigned a value during data extraction or assignment). If it is not known then ***TRUE*** is returned; otherwise it returns ***TRUE*** if the known value is zero; Otherwise ***FALSE*** is returned.

Syntax: boolean IsZeroOrNotKnown(TUPPLE Measure)

Parameters:

Measure The measure that the function determines whether it is not known or is zero.

Example:

```
IsZeroOrNotKnown(TaxableIncome)
```

Returned ***TRUE*** if no taxable income is known by the system or if the declared income is zero; otherwise ***FALSE*** is returned.

See also:

AllZeroOrNotKnown
IsEmpty
IsKnown
IsKnownAndNotZero
IsNotKnown

12.2.39 LeftStr (string s, double length)

Description: Returns the first *length* characters from the string (*s*). If *s* is shorter than *length* then the whole *s* is returned.

Syntax: string LeftStr(string s, double length)

Parameters:

s The string from which the first (left) characters is returned.

length The maximum number of characters returned.

Example:

```
LeftStr("Intracom IT-Services", 8)
```

Returns the text “Intracom”

12.2.40 LngToDbl (long value)

Description: Converts an integer value (*value*) into a decimal value.

Syntax: double LngToDbl (long value)

Parameters:

value The integer value that is converted into a decimal value.

Example:

```
LngToDbl(36)
```

Returns the decimal value 36.0

12.2.41 LngToStr (long value)

Description: Converts an integer value (*value*) into a string.

Syntax: string LngToStr (long value)

Parameters:

value The integer value that is converted into a string.

Example:

```
LngToStr(36)
```

Returns the string value "36"

12.2.42 LowerCase (string s)

Description: Returns the contents of the supplied string but with all cases converted into lower case.

The character sets supported depends on the setup of the actual server platform.

Syntax: string LowerCase(string s)

Parameters:

s The string which is converted into lower cases.

Example:

```
LowerCase("This is a text")
```

Returns the text "this is a text"

12.2.43 Maximum (double x_1, double x_2)

Description: Returns the maximum value of the two specified values (x_1 and x_2)

Syntax: double Maximum(double x_1, double x_2)

Parameters:

x_1 The value returned if it is greater than x_2.

x_2 The value returned if it is greater than x_1.

Example:

```
Maximum(3, 6)
```

Returns the value 6.

12.2.44 MaximumValue (TUPLE Measure, string DimensionName)

Description: Returns the maximum value of measure *Measure* for the dimension *DimensionName.*

Syntax: double MaximumValue(TUPLE *Measure*, string *DimensionName*)

Parameters:

Measure	The measure member from which maximum value will be derived from the specified dimension (*DimensionName*)**.**
DimensionName	The name of the dimension from which the maximum value is derived.

Example:

```
MaximumValue(TaxableIncome, "Years")
```

Returns the maximum taxable income found in the years.

12.2.45 Minimum (double x_1, double x_2)

Description: Returns the minimum value of the two specified values (x_1 and x_2)

Syntax: double Minimum(double x_1, double x_2)

Parameters:

x_1 The value returned if it is less than x_2.

x_2 The value returned if it is less than x_1.

Example:

```
Minimum(11, 3.14)
```

Returns the value 3.14.

12.2.46 MinimumValue (TUPLE Measure, string DimensionName)

Description: Returns the minimum value of measure *Measure* for the dimension *DimensionName.*

Syntax: double MinimumValue(TUPLE *Measure*, string *DimensionName*)

Parameters:

Measure	The measure member from which minimum value will be derived from the specified dimension (*DimensionName*)**.**
DimensionName	The name of the dimension from which the minimum value is derived.

Example:

```
MinimumValue(TaxableIncome, "Years")
```

Returns the minimum taxable income found in the years.

12.2.47 OnChanged (string key)

Description: Returns ***TRUE*** if the specified key (*key*) is different from the key specified last time the function was called; otherwise the function returns ***FALSE***. If it is the first time the function is called ***TRUE*** is also returned.

Syntax: boolean OnChanged(string key)

Parameters:

key The key that is compared to the key used in previous call.

Example:

```
OnChanged("MyKey")
```

Returns ***FALSE*** if the last call to OnChanged also was done with the key "MyKey"; Otherwise it returns ***TRUE***.

12.2.48 p (string ParameterTableName)

Description: Returns a parameter value from the specified parameter table (*ParameterTableName*) (for the taxpayers industry code).

Syntax: double p(string *ParameterTableName*)

Parameters:

ParameterTableName The name of the parameter table from which the value is returned.

Example:

```
p("SalesToIncomeRatio")
```

Returns the value stored in "SalesToIncomeRatio" parameter table for the tax payers industry code.

12.2.49 pa (string ParameterTableName)

Description: Parameter lookup function. Returns the average of the specified parameter value (*ParameterTableName*) within the years in CalendarYears dimension.

Syntax: double p(string *ParameterTableName*)

Parameters:

ParameterTableName The name of the parameter table from which the value is returned.

Example:

```
pa("SalesToIncomeRatio")
```

Returns the average value stored in "SalesToIncomeRatio" parameter table for the tax payers industry code.

12.2.50 py (string ParameterTableName, string Year)

Description: Returns a parameter value from the specified parameter table (*ParameterTableName*), for the year Year.

Syntax: double py(string *ParameterTableName*)

Parameters:

ParameterTableName The name of the parameter table from which the value is returned.

Example:

```
py("IncomeTaxRate", "2010")
```

Returns the income tax rate for the year 2010.

12.2.51 QualifiedAverage (TUPLE Measure, double Threshold, string DimensionName)

Description: Returns the average for values of measure (*Measure*), exceeding the threshold value (*Threshold*), for the dimension (*DimensionName*).

Syntax: double QualifiedAverage(TUPLE Measure, double Threshold, string DimensionName)

Parameters:

Measure	The measure member from which the total deviation will be derived**.**
Threshold	The threshold of the measures which are qualified for the average. Measures below this threshold will not be taken into account.
DimensionName	The name of the dimension from which the qualified average is derived.

Example:

```
QualifiedAverage(TaxableIncome,0,"Years")
```

Returns the qualified average of the positive taxable income over the years.

12.2.52 QualifiedCount (TUPLE Measure, double Threshold, string DimensionName)

Description: Returns the number of values for measure (*Measure*) that exceed the threshold value (*Threshold*), for the dimension (*DimensionName*).

Syntax: double QualifiedCount(TUPLE Measure, double Threshold, string DimensionName)

Parameters:

Measure	The measure member from which number is counted**.**
Threshold	The threshold of the measures which are qualified for the qualified count. Measures below this threshold will not be counted.
DimensionName	The name of the dimension from which the values are counted.

Example:

```
QualifiedCount(TaxableIncome,0,"Years")
```

Returns the count of measures of positive taxable income over the years.

12.2.53 randomsample (double rate)

Description: Returns ***TRUE*** by the likelihood specified by *rate*; and ***FALSE*** by the likelihood specified by 1 – *rate*. The function is based on a stochastic pseudo random function.

Syntax: boolean randomsample(double rate)

Parameters:

rate The probability by which ***TRUE*** is returned.

Example:

```
randomsample(0.2) == true
```

12.2.54 Ratio (double arg_1, double arg_2, double arg_3, double arg_4)

Description: Implements a safe division of $\frac{arg_1}{arg_2}$.

The result is calculated as follows:

If arg_2 is zero then arg_3 is returned.

If arg_1 or arg_2 (or both) are undefined (null) then arg_4 is returned.

Otherwise the value calculated as $\frac{arg_1}{arg_2}$ is returned.

If any errors occur then arg_4 is returned.

Syntax: double Ratio(arg_1, arg_2, arg_3, arg_4)

Parameters:

arg_1 The nominator of the division

arg_2 The denominator of the division

arg_3 The value returned if division by zero

arg_4 The value returned in case of errors or in case of arg_1 or arg_2 (or both) is undefined.

Example:

```
Ratio(3, 6 1, 2)
```

Returns the value 0.5.

```
Ratio(3, 0, 1, 2)
```

Returns the value 1.

12.2.55 RatioChange(TUPLE numerator, TUPLE denom, long year)

Description: Same as RatioIncrease.

12.2.56 RatioDrop (TUPLE numerator, TUPLE denom, long year)

Description: This function returns the negated value of *RatioIncrease*.

Returns the decrease (drop) in the ratio between the passed entries between a year and the following year.

The two passed coordinates must each have exactly one member from a period dimension. This function finds the next period on the same level for both coordinates, and calculates the ratio increase as follows:

$$\frac{numerator}{denom} - \frac{\{numerator, nextperiod\}}{\{denom, nextperiod\}}$$

The functions test for NULL or EMPTY conditions, and returns NULL or EMPTY if the ratio cannot be calculated without zero-division.

Note: an increase in the ratio will result in a negative result.

Syntax: double RatioDrop(TUPLE numerator, TUPLE denom, long year)

Parameters:

numerator The measure used as numerator.

denom The measure used as denominator.

year The number of years over which the ratio is derived.

Example:

```
RatioDrop(GrossReceipts, YearDebtSales, 0)
```

12.2.57 RatioIncrease(TUPLE numerator, TUPLE denom, long year)

Description: Returns the Increase in the ratio between the passed entries between a year and the following year.

The two passed coordinates must each have exactly one member from a period dimension. This function finds the next period on the same level for both coordinates, and calculates the ratio increase as follows:

$$\frac{\{numerator, nextperiod\}}{\{denom, nextperiod\}} - \frac{numerator}{denom}$$

The functions test for NULL or EMPTY conditions, and returns NULL or EMPTY if the ratio cannot be calculated without zero-division.

Note: a decrease in the ratio will result in a negative result.

Syntax: double RatioIncrease(TUPLE numerator, TUPLE denom, long year)

Parameters:

numerator The measure used as numerator.

denom The measure used as denominator.

year The number of years over which the ratio is derived.

Example:

```
RatioIncrease(GrossReceipts, YearDebtSales, 0)
```

12.2.58 RightStr (string s, double length)

Description: Returns the last *length* characters from the string (*s*). If *s* is shorter than *length* then the whole *s* is returned.

Syntax: string RightStr(string s, double length)

Parameters:

s The string from which the last (right) characters is returned.

length The maximum number of characters returned.

Example:

```
RightStr("Intracom IT-Services", 8)
```

Returns the text “Services”

12.2.59 Roundsum (double value)

Description: Returns the number of continuous zeros in the specified value (*value*) taken from the right.

Syntax: double Roundsum(VARIENT value)

Parameters:

value The value which trailing zeros are counted.

Example:

```
Roundsum(9000)
```

Returns the value 3.0

12.2.60 SoundsLike (string s_1, string s_2)

Description: Compares two strings (s_1 and s_2) using SOUNDEX (see http://en.wikipedia.org/wiki/Soundex). I.e. by converting each of the strings by using SOUNDEX and then string comparing the two results. Hereby a simple pattern comparison can be made.

Syntax: boolean SoundsLike(string s1, string s2)

Parameters:

s_1 The string which SOUNDEX value is compared to the SOUNDEX value of s_2.

s_2 The string which SOUNDEX value is compared to the SOUNDEX value of s_1.

Example:

```
SoundsLike("Intracom", "Intrekom")
```

Returns ***TRUE*** because the two words “Intracom” and “Intrekom” sounds a lot like each other.

```
SoundsLike("Intracom", "Entrecote")
```

Returns ***FALSE*** because the two words “Intracom” and “Entrecote” sound too different.

12.2.61 StartDate (string period)

Description: Gets the start date from a period string (format *YYYYMMDD-YYYYMMDD*) and return it as an integer in the format *YYYYMMDD*.

Syntax: long StartDate(string period)

Parameters:

period The string that is interpreted as a period and which star date is returned as an integer.

Example:

```
StartDate("19660714-20100505")
```

Returns the integer value 19660714

12.2.62 StrToDbl (string s)

Description: Converts (if possible) the specified string into a decimal number.

Syntax: double StrToDbl(string s)

Parameters:

s The string is converted into a decimal number.

Example:

```
StrToDbl("7.913")
```

Returns the double value 7.913

12.2.63 StrToLng (string s)

Description: Converts (if possible) the specified string into an integer.

Syntax: long StrToLng(string s)

Parameters:

s The string is converted into an integer.

Example:

```
StrToLng("7.913")
```

Returns the integer number 7

12.2.64 SubStr (string s, double pos, double length)

Description: Returns a substring as the *length* characters from the string (*s*) starting at position *pos*.

Syntax: string SubStr(string s, double pos, double length)

Parameters:

s The string from which the substring is returned.

pos The position of the first characters in the returned substring, where 1 is the first character in the string (*s*).

length The maximum number of characters returned.

Example:

```
SubStr("Intracom IT-Services", 10, 2)
```

Returns the text "IT"

12.2.65 Today ()

Description: Returns the current date as a string in the format *YYYYMMDD*.

Syntax: string Today()

Example:

```
Today()
```

12.2.66 TotalDeviation (TUPLE Measure)

Description: Returns the total deviation for the measure (*Measure*), based on the members on the period dimension.

Syntax: DOUBLE TotalDeviation(TUPLE Measure)

Parameters:

Measure The measure member from which the total deviation will be derived**.**

Example:

```
TotalDeviation(TaxableIncome)
```

Returns the total deviation of the taxable income over the period dimension.

12.2.67 TotalDeviationPct (TUPLE Measure)

Description: Returns the total deviation percentage for the measure (*Measure*), based on the members on the period dimension.

Syntax: double TotalDeviation(TUPLE Measure)

Parameters:

Measure The measure member from which the total deviation percentage will be derived**.**

Example:

```
TotalDeviationPct(TaxableIncome)
```

Returns the total deviation percentage of the taxable income over the period dimension.

12.2.68 TrimString (string s)

Description: Returns a copy of the specified string (*s*) but where all trailing spaces, tabs and other whitespace characters have been removed.

Syntax: string TrimString(string s)

Parameters:

s The string which is trimmed.

Example:

```
TrimString("Intracom    ")
```

Returns the text “Intracom”

12.2.69 UpperCase (string s)

Description: Returns the contents of the supplied string but with all cases converted into upper case.

The character sets supported depends on the setup of the actual server platform.

Syntax: string UpperCase(string s)

Parameters:

s The string which is converted into upper cases.

Example:

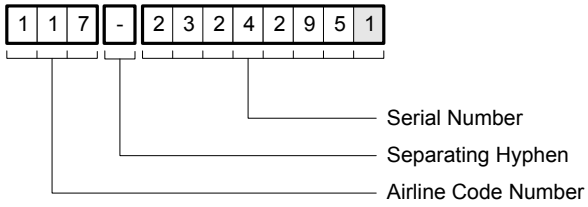
```
UpperCase("This is a text")
```

Returns the text “THIS IS A TEXT”

12.2.70 ValidAirWaybillNumber(string awbn)

Description: Validates the Air Waybill Number specified by *awbn*. If *awbn* specifies a valid airway bill number then the function returns ***TRUE***; otherwise the function returns ***FALSE***.

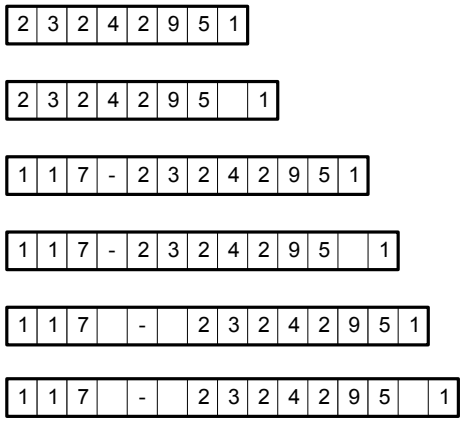
As shown in the figure below the Air Waybill Number consists of the following parts:



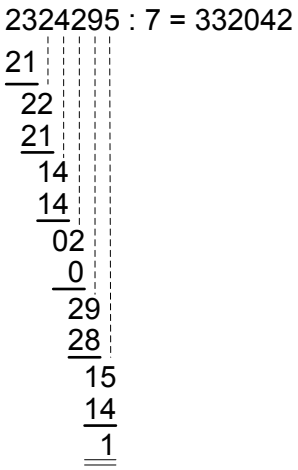
The Air Waybill Number

1) Airline Code Number
2) Separating Hyphen
3) Serial Number

The validation is made on the Serial Number only; the function however is implemented to handle both complete Air Waybill Numbers and the Serial Number part only. This is done by letting the function look at the last 8 digits of the filtered *awbn*-parameter only – (Filtered means that if separating white spaces exists they will be ignored) – hereby the following *awbn*'s will be treated equally as the Serial Number 23242951:



A Serial Number is validated by dividing the first 7 digits by 7 (in the example this will be 2324295). For a valid Serial Number the remainder is the same number as the last digit (in the example this is 1). In such case the Serial Number and thereby the whole Air Waybill Number is considered valid and the function returns ***TRUE***; otherwise it is considered invalid and the function returns ***FALSE***.



Syntax: boolean ValidAirWaybillNumber(string awbn)

Parameters:

awbn The string representing the Air Waybill Number that is validated.

Example:

```
ValidAirWaybillNumber("117-2324295 1")
```

Returns the ***TRUE*** because 117-2324295 1 is a valid Air Waybill Number.

12.3 Reference sections

12.3.1 Numeric Display Formats

The display format for numbers provides a way of defining different settings of how numbers such as amounts, reals and integers should be displayed. The format (see below for formal syntax) can have one to four sections with semicolons (;) as the list separator. Each section contains the display format for a different type of number conditions:

Section	Description
First	The format for positive numbers.
Second	The format for negative numbers.
Third	The format for zero value numbers.
Fourth	The format for Unknown value. *Note that the use of this format is not currently implemented!

If you use multiple sections but don't specify a format for each section, entries for which there is no format will default to the formatting of the first section.

Within a formatting section the amount placeholder (%N) may be present. The placeholder specifies where in the format the number will be displayed. Any literal characters before the placeholder will be displayed as prefix to the number and any literal characters after the placeholder will be displayed after the number. If the amount placeholder is not present there is no suffix. The amount placeholder is generally only needed if you have literal characters in the suffix.

Just before the number placeholder you may specify one or more formatting flags, that each specifies a special display setting. See "Formal Format Display Syntax" section below for a more formal definition of the format.

If an invalid display format is specified the displayed text will be "###" indicating an error.

12.3.1.1 Precision Flag

Flag Syntax	"%P" <Digit> * Note: The current implementation pads with trailing zeros accordingly to precision - see %TZ flag.
Description	Specifies the precision of the amount. This corresponds to the maximum number of significant digits after the decimal point character. If precision is set to 0 the decimal point character will not be used. The amount will be rounded if necessary thus a amount of 1.238 with a specified precision of 2 will be rounded to 1.24. Note that an amount of 1.2 with a specified precision of 2 is 1.2. To print trailing zeros use %TZ flag, e.g. 1.20.
Default setting	The default precision is taken from regional settings - see below.

12.3.1.2 Digit Grouping

Flag Syntax	"%G" [<Digit>* ["'" <Char> "'"]] * Note: Currently only one digit for group sizes are implemented.
Description	Specifies the digit grouping and the digit grouping separator character. This flag will be ignored if scientific notation is specified. The digit grouping is specified by a series of digits. Each digit specifies the number of digits in a grouping starting from right to left (from the decimal point character). The last digit specifies the grouping for subsequent groups.

The 0 digit may only be used as the last digit.

Example: Thus a specification of 324 will group first with 3 digits and then with 2 digits followed by subsequent groups of 4 digits. Example "111 22 3333 4444"

Default setting The default grouping units and the default grouping separator character is taken from the regional settings - see below. Grouping is not used unless %G is specified.

12.3.1.3 Decimal Point Character

Flag Syntax "%D" """ <Char> """

Description Specifies the decimal point character. The decimal point character will not be displayed if precision is set to 0.

Default setting The default decimal point character is taken from the regional settings - see below.

12.3.1.4 Scientific notation

Flag Syntax "%E" [<Sign>]
Note: Currently implementation ignores <Sign> and forces the sign on.

Description Specifies that the number should be displayed with scientific notation. The sign specifies whether + or - after the 'E' should be forced/removed. The exponent will always be displayed with 3 digits padded if necessary with zeros.

Note: The exponent is always prefixed with a capital 'E'. Currently it's not possible to control whether the exponent is prefixed with an 'E' or 'e'.

Default setting Do not display with forced sign. Scientific notation is not used unless %E is specified.

12.3.1.5 Force or remove sign

Flag Syntax "%+" or "%-"

Description In positive number format specification this forces the sign to be presented. In negative number format specifications it removes the sign to be presented.

Note: Be careful with only specifying a positive number format using this flag. When using this flag you should always supply a negative number format, unless you're sure that the sign have no importance in negative numbers.

Default setting	No forced sign in positive format and no remove sign in negative format.

12.3.1.6 Trailing zeros

Flag Syntax	"%TZ" *Note: Current implementation assumes this flag - always.
Description	Specifies that trailing zeros should be padded accordingly to the precision. Thus an amount of 1.2 with precision 2 and trailing zeros will be padded with an extra zero, thus becoming "1.20".
Default setting	No trailing zeros.

12.3.1.7 Leading zeros

Flag Syntax	"%LZ" <Digit> *Note: Currently not implemented.
Description	Specifies that leading zeros should be padded in front of the first digit in the amount. The digit specifies the maximum number of zeros to pad with. Thus an amount of 1.2 with a leading zero specification of 3 will be displayed as "001.2".
Default setting	No leading zeros.

12.3.1.8 Multiplication

Flag Syntax	"%M" <Integer>
Description	Specifies that the value should be multiplied with an integer value as specified. The integer value should be different from zero. This can be used in e.g. percent values to show 0.5 as 50 (if integer value is specified as 100).
Default setting	Does not multiply (integer equals zero).

12.3.1.9 Formal Syntax

<Display format string> ::=
<Positive number format>
[";" <*Negative number format*> [";" <*Zero number format*>
[";" <Unknown number format>]]]

<Positive number format> ::=
    <Format>
<Negative number format> ::=
    <Format>
<Zero number format> ::=
    <Format>
<Unknown number format> ::=
    <Format>

Format

<Format> ::=
    <Prefix> <Flag>* [<Number placeholder> <Suffix>]
<Prefix> ::=
    <Characters>
<Suffix> ::=
    <Characters>
<Number placeholder> ::=
    "%N"

Format Flags

<Flag> ::=
    <Precision> | <Decimal point char> | <Grouping> | <Scientific notation> |
    <Force sign>
<Precision> ::=
    "%P" <Digit>
<Decimal point char> ::=
    "%D" <Single quote> <Char> <Single quote>
<Grouping> ::=
    "%G" [<Digit> [<Single quote> <Char> <Single quote>]]
<Scientific notation> ::=
    "%E" [<Sign>]
<Force sign> ::=
    "%+" | "%-"

12.3.1.10 The Default Settings

The default values used in some of the flags are pr. default taken from the Numbers-tab from the 'Control Panel | Regional Options' dialog within your Windows operating system. This does not apply to UNIX installations.

12.3.1.11 Examples

This section describes some Number display formats. You should note that some of the examples below the result may depend on the current settings within the regional options dialog, since these are used per default. Also in the examples below it is assumed that the "English (United States)" user locale is selected. Thus

the default precision is 2, the default decimal point character is '.', the default grouping character is ',' and the default grouping is ''

Display Format	**Sample amount**	**Will be displayed as (excl. quotes)**	**Comments**
"" or "%N"	123456.789	"123457"	The most simple display format. No prefix, suffix or formatting flags specified.
"" or "%N"	-123456.789	"-123457"	
"Amt= %N."	123456.789	"Amt=123457."	Use of literal characters in prefix and suffix. Prefix equals "Amt= " and suffix equals ".".
";-;Null;Unknown"	123456.789	"123457"	
";-;Null;Unknown"	-123456.789	"-123457"	Need minus character as prefix.
";-;Null;Unknown"	0	"Null"	Null is displayed literally.
";-;Null;Unknown"	Unknown	"Unknown"	Unknown is displayed literally.
"%+;%-"	123456.789	"+123457"	Use of force sign. It doesn't matter whether %+ or %- are used.
"%+;(%-)"	-123456.789	"(123457)"	Use of force sign. It doesn't matter whether %+ or %- are used.
"%P2%TZ"	123456.789	"123456.79"	
"%P2%TZ"	123456.7	"123456.70"	
"%G"	123456.789	"123 457"	Use of default grouping of 3 digits and default grouping separator character " " (space).

Display Format	Sample amount	Will be displayed as (excl. quotes)	Comments
"%G32'.'%D','%P2"	123456.789	"1.23.456,79"	
"%G32'.'%D','%P2"	123456.7	"1.23.456,7"	
"%G32'.'%D','%P2%TZ"	123456.7	"1.23.456,70"	
"%G3','%D'.'%P2%TZ; (%-G3','%D'.'%P2%TZ); %P2%TZ"	123456.7	"123,456.70"	
"%G3','%D'.'%P2%TZ; (%-%G3','%D'.'%P2%TZ); %P2%TZ"	-123456.7	"(123,456.70)"	
"%G3','%D'.'%P2%TZ; (%-%G3','%D'.'%P2%TZ); %P2%TZ"	0	"0.00"	
"%E"	123456.789	"1E005"	If precision is 0 within regional settings.
"%E%P3"	123456.789	"1.235E005"	
"%E+%P3"	123456.789	"1.234E+005"	
"%E+%P3"	-123456.789	"-1.234E+005"	
"%E%P3"	0.123456789	"1.235E-001"	
"%E%P3%+"	0.123456789	"+1.235E-001"	
"%E%P3%+"	-0.123456789	"-1.235E-001"	
"%LZ2"	-0.123456789	"-00"	
"%LZ2"	1.23456789	"01"	
"%P2%LZ2"	1.23456789	"01.23"	
"%P2%LZ2"	1.2	"01.2"	
"%P2%LZ2%TZ"	1.2	"01.20"	
"%P2%LZ2%TZ"	1.234	"01.23"	

Display Format	Sample amount	Will be displayed as (excl. quotes)	Comments
"%X"	1.234	"###"	Error in display format.
"%M100 %%"	0.25	"25 %"	

Index

C
Columns
  hiding, 28
  moving, 25
Compliant List
  add a trader, 102
  delete a trader, 104
  edit a trader, 103
  printing, 104
  working with, 102
Configuration element
  closing, 29
  Compliant List, 102
  copying, 23
  deleting, 23
  displaying, 29
  keyword, 33
  printing, 23
  Random Selection, 112
  Result Table, 105
  revisions, 10
  Risk Objects, 108
  selecting, opening, 22
Conventions
  Mouse operation, 7
  Notation, 7
D
Data Area, 28
  closing configuration elements, 29
  displaying configuration elements, 29
Declaration
  segments, 32
  sub-tab, 55
  types, 30
E
ESKORT Clearance Risk Analysis System
  Modules, 9
  Risk Analysis Process, 9
ESKORT Clearance Workbench
  Conformity, 8
F
Filtering
  of Risk Profiles, 60
  using criteria, 61
  using ID number, 62
Formalized Expression View, 69
  deleting, undoing and redoing conditions, 84
  Operators, 85
  validating rules, 84
  writing conditions, 70
H
Hyperlink, 34
  create, 35
  edit, 35
  open, 35
I
Indicators, 84
M
Main Areas, 19
  Data Area, 28
  Filter Area, 19
  Table Area, 21
Main menus, 13
  Edit menu, 15
  File menu, 14
  Help menu, 16
MDC Language, 128
  Function reference, 134
  Grammar, 128
  terminal symbols, 133
Mouse operation
  Conventions, 7
  highlight (select), 7
N
Numeric Display Formats, 167
O
Operators, 85
P
Partial Expression View, 66
  colums, 66
  deleting, undoing and redoing conditions, 69
  setting conditions, 67
Printing
  of Compliant Lists, 104
  of Random Selections, 113
  of Result Tables, 107
  of Risk Objects, 111
  of Risk Profiles, 63

R
Random Selection
deleting, 113
editing, 112
new, 112
printing, 113
working with, 112
Result Table
deleting, 106
editing, 106
new, 105
printing, 107
working with, 105
Risk Analysis Configuration
configuration elements, 9
introduction, 9
Risk Analysis Process
Introduction, 9
Risk Object
deleting, 110
editing, 110
new Commodity Risk Object, 108
new Country Risk Object, 109
new Procedure Risk Object, 108
new Trader Risk Object, 108
printing, 111
Risk Objects
working with, 108
Risk profile
control request, 57
Description, 56
editing, 60
examination instructions, 58
filtering, 60
Indicators, 84
new, 38
printing, 63
saving, 60
selecting inspection type, 58
selecting report back fields, 59
sub-tab, 57
using rule buttons, 65
working with, 37
working with Rules, 63
S
Sorting
Table Area Contents. *See* Table Area
T
Table Area
hiding columns, 28
sorting contents, 26
U
User Interface, 11
main areas, 19
main menus, 13
resizing, 12
screen layout, 11
the toolbar, 17
V
Views
display buttons, 65
dividing bars, 64
re-sizing, 64
re-sizing arrows, 64
the Formalized Expression view, 69
the Partial Expression View, 66
W
Wildcard, 36

□